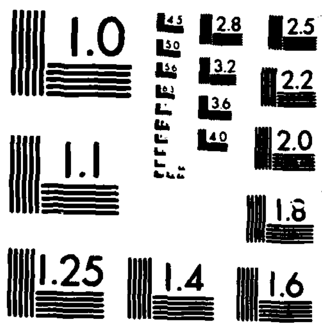
AD-A163 944 A FIGHTER PILOT'S INTELLIGENT AIDE FOR TACTICAL MISSION 1/2
PLANNING(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB
OH SCHOOL OF ENGINEERING R B BAHNIJ DEC 85
UNCLASSIFIED AFIT/GCS/ENG/85D-1 F/G 6/4 NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

A FIGHTER PILOT'S INTELLIGENT AIDE FOR TACTICAL MISSION PLANNING

THESIS

Robert B. Bahnij
Major, USAF

AFIT/GCS/ENG/85D-1

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GCS/ENG/85-1

A FIGHTER PILOT'S INTELLIGENT AIDE FOR

TACTICAL MISSION PLANNING

THESIS

Robert B. Bahnij
Major, USAF

AFIT/GCS/ENG/85D-1

Approved for public release; distribution unlimited

AFIT/GCS/ENG/85D-1

A FIGHTER PILOT'S INTELLIGENT AIDE FOR TACTICAL MISSION PLANNING

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Computer Systems

Robert B. Bahnij, B.S., M.S.
Major, USAF

December 1985

Preface

The purpose of this research was to fuse my extensive fighter instructor pilot experience with a computer engineering artificial intelligence education to design, implement, and evaluate a prototype tactical mission planning system. This system would not only make life easier for the fighter pilot, but also for the commercial system designer.

I would like to thank the United States Air Force for giving me the opportunity to temporarily leave the cockpit and pursue an advanced engineering education.

I thank Steve Cross, the Meta-guide, for creating a research and learning environment for exploiting the strengths of both student and machine, creating a force multiplier for addressing current and future Air Force operational needs. I thank Lt Col Ron Morishige, the Pilot's Associate Program Manager, for his friendship and support, especially the TDY funding.

If I had to credit any single individual for helping me get or maintain an academic 'clue,' especially in the Theory of Computation course sequence, it would have to be Doug Norman. Doug has the gift for transforming the 'muddy delta waters' into a 'clear mountain spring.' It makes drinking from the AFIT 'fire hose' a more palatable experience.

Thanks are in order to John Mitchiner and Laurie Phillips, of Sandia Labs, for allowing me to use their

algorithm, which served as the basis for the terrain profile views and SAM threat detection system capabilities.

Three special individuals, all first class lisp hackers and computer wizards, are Jim Loftus (Hoser senior), Neal Feinberg (Hoser junior), and Kalman Reti, from the Symbolics Research Center in Cambridge MA. There was never a time, day or night, that I could not call on any one of them for advice and not get the 'answer.' Thanks for leading me down the Hacker's path to Steve's, where Flavors was created. Their unselfish friendship is truly appreciated.

However, the greatest debt is owed my wife Cindy and daughter Nicole. Their continuous love, support, sacrifices, patience, and strength have made this educational endeavor truly successful. I dedicate this thesis to the two brightest stars in my universe, Cindy and Nicole, for without their warm and loving light, the successful completion of this thesis, as well as the entire AFIT graduate engineering program would not have been possible.

Robert B. Bahnij

Table of Contents

List of Figures

Figure | Page

Figure		Page

AFIT/GCS/ENG/85D-1

Abstract

A working tactical mission planning prototype is described that automates many of the labor intensive, computationally demanding tasks now associated with tactical mission planning. This prototype focuses the pilot's attention on the higher level aspects of the mission, such as, contingency exploration; simultaneously generating the immediate product, a refined mission plan. It also exploits the strengths of both man and machine to overcome the shortcomings of each, producing a 'win-win' situation. The interactive use of this prototype has the capability to synergistically increase tactical mission situation awareness, on which the pilot will base actual in-flight critical decisions.

The present approach to tactical mission planning has several disadvantages. The pilot must concentrate on isolated subtasks. For instance, he must manually determine mission relevant navigational coordinates from maps. He must then type the coordinates into a hand-held calculator or the squadron's PC to determine critical parameters, such as, leg length and fuel used. The plan is refined iteratively. Artificial intelligence techniques can off-load

many of these low level tasks and help the pilot deal with mission complexities. This not only "takes the drudgery" out of mission planning, it improves the pilot's overall mission situation awareness.

This prototype knowledge-based system, designed and implemented by a fighter instructor pilot, overcomes present disadvantages and provides several new capabilities. Examples of new capabilities include: identification and proposed resolution of constraint violations, such as, computer generated advice on threat avoidance options and pilot specification of three dimensional terrain profiles of proposed flight paths.

The research demonstrates that a knowledge-based programming language facilitates system design by domain experts. This language will permit squadron pilots, the end users, to define commercial system requirements. The thesis will describe this system and discuss a preliminary evaluation by Air Force pilots.

A FIGHTER PILOT'S INTELLIGENT AIDE FOR TACTICAL MISSION PLANNING

I. Introduction

Background

"To meet the challenge of certain critical problems in defense," the Defense Research Projects Agency (DARPA) initiated an important new program, and in October 1983, published a technical report outlining the Strategic Computing Program (DARPA, 1983:i). The overall goal of the Strategic Computing Program (SCP) is "to provide the United States with a broad line of machine intelligence technology, and to demonstrate applications of the technology to critical problems in defense" (DARPA, 1983:ii). The sophistication, speed, and number of current weapon systems, both threat and friendly, create an exponential growth in the decision making process. Filtering, assessing, and focusing the tremendous amount of data, and managing the accelerated information flow, are tasks required to produce complex decisions in a timely manner. The physical environment in which these decisions occur is wrought with noise, vibration, and in the case of fighter aircraft, violent maneuvers, with associated high "g" forces. Uncertainty, especially, the unpredictable nature of adversary actions creates a greater affinity for information. These are but a few examples of the critical problems in defense that require immediate resolution. The

top "key areas of advances, that can be leveraged to produce high-performance machine intelligence," identified as the base for the SCP was "Expert Systems: Codifying and mechanizing practical knowledge, common sense, and expert knowledge" (DARPA, 1983:ii).

DARPA is sponsoring the development of four military applications programs:

1. An autonomous vehicle program (U.S. Army).
2. A carrier battle group management system (U.S. Navy).
3. An air/land battle management system (U.S. Air Force and U.S. Army).
4. A pilot's associate program (U.S. Air Force).

The Air Force Wright Aeronautical Laboratories (AFWAL) at Wright-Patterson AFB was selected as DARPA's agent for the Pilot's Associate (PA) Program. The pilot's associate can be viewed as an intelligent system that assists the fighter pilot, both on the ground and in the air, to off-load lower-level tasks and perform special functions enabling the pilot to maintain situation awareness and focus on higher-level strategic and tactical objectives (DARPA, 1983:24). The Pilot's Associate Program office has designated five major functional areas for implementation, applying both conventional algorithmic techniques and artificial intelligence technology: situation assessment, systems monitoring, tactical planning, mission planning, and pilot/vehicle interface (AFWAL, 1985:1).

The situation assessment function will assist the pilot with characterizing and evaluating the effect of threats,

weather, and aircraft status on mission objectives. The tactical planner will integrate the inputs from the Situation Assessment (SA), System Status Monitor (SSM), and the Mission Planner (MP) functions to recommend offensive and defensive actions. The mission planner helps the pilot execute and/or revise the mission plan by providing mission options consistent with the constraints input from the SA, TP, and pilot. The system status monitor will advise the pilot of current or anticipated malfunctions and the complex implications of each single or multiple malfunction. The SSM will monitor the status of on-board, as well as, external resources. The pilot/vehicle interface (PVI) will combine advanced controls, displays, and interface devices to establish the environment for the communication link between the pilot and the Pilot's Associate.

Problem Definition

Tactical mission planning in operational fighter units is excessively time consuming and labor intensive. This situation is incompatible with the rapid deployment force (RDF) concept of the Readiness Command, which is ready, with short notice, to deploy and conduct tactical warfare anywhere in the world. This process is also characterized by physically distributed knowledge sources (i.e., the weatherman and the intelligence officer are located in different areas of the base). The mission planners (fighter aircrews) must gather, and integrate, many pieces of essential

information (i.e., target data, weapons effects and appropriate weapons delivery options and tactics, threat analysis, current and forecast target/enroute weather, low level route terrain analysis, force structure). The planners must deal with these, and many more, lower level constraints; and construct a plan that achieves mission objectives in a safe manner. Many lower level constraints and processes can be represented and modeled using artificial intelligence techniques. The research will define, design, implement, and evaluate a knowledge-based prototype interactive tactical mission planning system, to be used by fighter pilots in the squadron, prior to the actual flight. The synergistic benefit of increasing pilot situation awareness while producing a refined mission plan will be presented. This prototype can be used as a vehicle to help domain users define delivery system requirements.

Scope

The eight month project does not permit the realization of a fully operational Tactical Mission Planning System. It "still requires at least five man-years to develop a system that begins to be robust" (Davis, 1982:10). Thus, the project will not develop the many interface capabilities required to keep the planning system knowledge base current (i.e. weather and threat updates). Much of the natural language, explanation, and justification features of an knowledge-based system will not be supported. A basic user

interface, limited to menus and keywords, will be implemented.

Due to the classified nature of real world information, all enemy and friendly combat capabilities will not be used. The data representing mission scenarios (tactics) and weapon systems' capabilities will be older, unclassified, information published in numerous weapons review journals. Due to the limited allocated time, approximately ten weeks, to code the software, only the mission planning phase from takeoff to the IP will be implemented.

Assumptions

We assume all required interfaces can be implemented in the full production model. Real world classified information will replace the present unclassified database, once a tempest approved computer system is chosen to run this program. Tempest approval entails testing and approving a particular computer system to process classified data.

It is assumed that the reader is familiar, but not fluent, with the lisp programming language, the Symbolics programming environment, and KEE knowledge engineering tool. This thesis will assume the reader has an introductory level understanding of Artificial Intelligence techniques and concepts, such as, knowledge-based systems, production systems, object-oriented programming, and knowledge representation techniques.

Thesis Objectives and Approach

The goals of this ~~research~~ thesis are to design, implement, and evaluate a prototype tactical mission planning system.

The system design will be based on a thorough analysis of the present operational tactical mission planning process and applying state of the art artificial intelligence. A baseline design concept will be specified. Additionally, requirements for a knowledge-based language to support the design implementation will be presented.

The prototype will be implemented in a knowledge-based language customized for the tactical mission planning process. The language will be based on Zetalisp and Kee, release 2.1. It will facilitate the prototype development by a domain expert. The domain expert will use the language to represent mission planning knowledge.

The prototype will be evaluated by Air Force pilots. The purpose of the evaluation is two fold. Design is an iterative process. It will be shown that a baseline prototype is a crucial first step in allowing operational personnel to define the system requirement. Results of the evaluation will be incorporated into the prototype. A measure of design success will be how rapidly user requirements can be incorporated. The evaluation will also provide some valuable insight into the use of such a system in the tactical arena and the potential increases in pilot situation awareness.

Materials and Equipment

The prototype knowledge-based system was implemented in Zetalisp on a Symbolics 3670 LISP Machine, also using their high resolution eight bit color monitor. The KEE (Knowledge Engineering Environment) Software Development System, release 2.1, by IntelliCorp, was used as the building tool for the tactical mission planner.

Overview of the Thesis

Chapter two will introduce the reader to the current tactical mission planning process. The major disadvantages of current planning will be highlighted. Chapter three will discuss recent related work. Knowledge-based planning systems will be emphasized, especially military systems. Chapter four will construct the conceptual framework for the design of the proposed prototype. The important major concept of pilot situation awareness will be presented in detail. The author considers this section essential reading for understanding the complex nature of tactical aviation. Chapter five will describe the working prototype by actually guiding the reader through the mission planning process. Chapter six will summarize the research, highlighting the important contributions and conclusions. Recommendations for further research and extensions will be offered.

II. Current Tactical Mission Planning

Introduction To USAF Fighter Missions

"The mission of tactical air power is to deter the enemy from attacking, and should deterrence fail, to conduct war at the level of intensity and effectiveness needed to win" (TACM 2-1, 1978: 1-1). The primary missions, involving fighter aircraft of the Tactical Air Force (TAF) are: Counter Air (CA), Air Interdiction (INT), and Close Air Support (CAS). Counter Air missions are further divided into Offensive Counter Air (OCA) and Defensive Counter Air (DCA) missions. Figure 1, page II-2, graphically portrays the integration of tactical airpower (TACM 2-1, 1978: 1-5).

The major objective of Offensive Counter Air is to gain, and maintain, "air superiority." Air superiority, control of the airspace, is essential for the successful exploitation of the mobility capability of surface operations, (that is, Army, Navy, and Marines.) Some representative mission types are: Attack, Fighter Sweep, Combat Air Patrol (CAP), and Air Escort. The objective of the Attack mission is to destroy resources that contribute to enemy air superiority. A strike against an enemy airfield and petroleum, oil, and lubrication (POL) storages, is an example of an OCA mission. Fighter Sweeps are used to sanitize an area from air threat prior to the approach of the strike package (the surface attack configured aircraft.) CAP missions also engage and destroy enemy air forces. Air escort missions

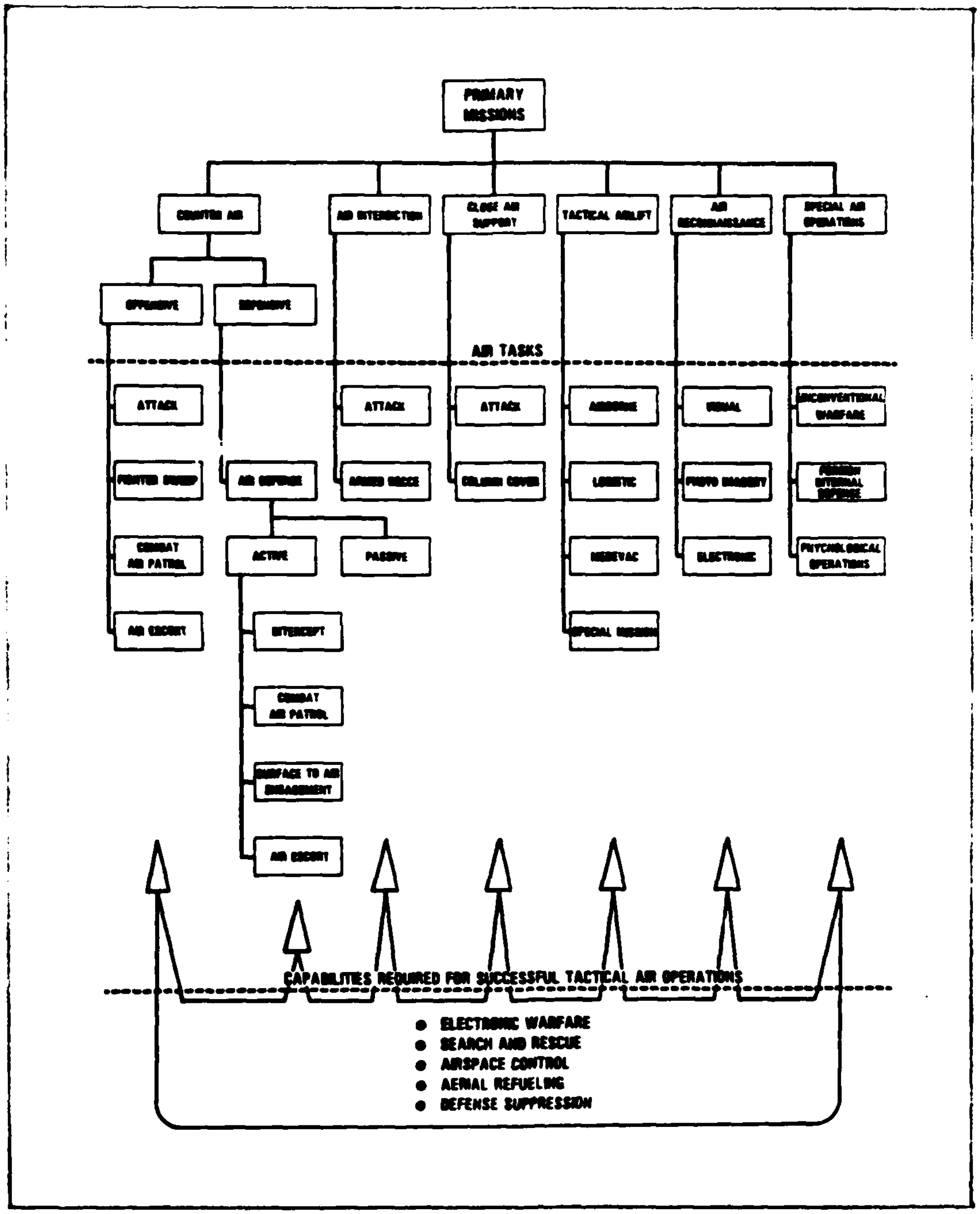


Figure 1. The Integration of Tactical Air Operations

are tasked with protecting the strike package from enemy air attacks.

Defensive Counter Air, as the title implies, is a defensive or reactionary mission. DCA missions are typically generated from an alert status, both on the ground and in the air. Air defense scrambles (minimum time from warning to launch) and air intercepts are the normal DCA type scenarios.

Air Interdiction mission objectives are to delay, neutralize, or destroy the military potential of the enemy before it can be effectively used against our own forces. Preplanned attacks against enemy targets behind enemy lines and armed reconnaissance (RECCE) missions are representative of Air Interdiction role.

Close Air Support (CAS) is used in direct support of ground forces. Targets are seldom known prior to takeoff, and usually fighters are briefed on their targets minutes before they start their attack.

Air Interdiction and Offensive Counter Air missions require extensive preplanning, coordination, and communication between mission elements. The OCA mission of an enemy airfield attack will be used as the vehicle to demonstrate the current process of tactical mission planning. Dynamic mission factors, such as, changing threats or area weather, cause a combinatorial explosion of options facing the planner. Although there can never be a standard tactical

mission plan, representative times and numbers are used in the mission planning examples presented in this work.

Overview of Mission Planning Process

The Air Tasking Order (ATO), generated by higher headquarters, starts the mission planning process at the base (Wing) level. The ATO assigns unit specific missions. The ATO is commonly referred to as the "FRAG", meaning Fragmentary Order. Although the ATO is not actually synonymous with the frag, within the scope of this thesis they will be used interchangeably. A frag is a unit specific, that is, squadron, subset of the ATO. The ATO addresses multiple units.

The following are some of the essential elements of information contained in the ATO; target identification and location, time the bombs need to be on target (TOT), unit assigned, type and number of aircraft, aircraft weapons configuration, the type, number, and callsign of all other support players (such as, reconnaissance, air escort, defense suppression (wild weasels), tankers, AWACS, electronic jammers, Army, Navy, Marines). This thesis will discuss the primary fighter/attack type aircraft. Thus, the ATO not only initiates the planning process, but also imposes the major high level constraints, time on target (TOT) and aircraft configuration. The ATO defines the Standard Configuration Load (SCL), from which the aircraft weight, the type and number of weapons (munitions) to carry, and fuel

requirements are determined. The SCL also provides the data required to determine the aircraft fuel flow. These data establish constraints on how far and fast the fighter can fly.

The ATO is transmitted to the base command post (location of the battle staff) via secure communication lines. The ATO is usually received early in the morning, (for example, 0200 to 0400) and details the preplanned missions for that morning. The message is deciphered, and after the battle staff reviews it, the ATO is hand carried to the "FRAG" break out team. The frag team consists of representatives from: maintenance, munitions (weapons), and operations (flyers). They extract relevant information and start to form an overall plan. These planning activities schedule the appropriate resources with the proper configuration to meet mission requirements, within a limited time period. After the ATO is dissected ("shredded") and reassembled in a relevant format, the frag is passed to the mission planning cell (an overall mission commander has been assigned the previous day: he has already prepared a generic plan that may or may not be useful for this particular tasking).

Next, a mass briefing is held to inform selected pilots the high level details of the unit's tasking. The pilots will be briefed with current intelligence, such as, war status, current and forecast weather (home base, enroute, and target), and the constraining details of the ATO.

Finally, the mission commander, along with other flight leaders, starts to put together the high level plan, paying particular attention to the interfacing issues, coordinating mutual support and deconflicting time and space requirements. Appendix A contains a representative "Combat Mission Planning Checklist" used in the 310th TFTS at Luke AFB as a guide to mission planning. The 310th F-16 training squadron is where the author was assigned as instructor pilot prior to his AFIT assignment.

After the high level details, assigning times and role responsibilities of all participating flights, of the overall mission have been integrated into one plan (the "Strike Package"), the individual flight mission planning takes place. Preplanned OCA missions (massive strike packages) generally involve numerous aircrafts. The strike package may number 30 to 50 aircraft composed of:

- Bombers	F-4, F-111, F16
- Defense Suppression	F-4G (wild weasels)
- Escort/Sweep	F-15, F-16, F-14 (Navy)
- Reconnaissance	RF-4C
- GCI	AWACS
- Tankers (Air Refueling)	KC-135, KC-10
- Electronic Jammers	EF-111, EA-6 (Navy)

The flight (lowest level mission element) will prepare and brief its proposed plan. After the individual flight briefs, the pilots will get final weather and intelligence updates as they are "suiting up" (putting on their flight gear). The pilots will "Step" (depart the squadron building enroute to the airplanes) as a flight, and begin the ground

operation portion of the mission. The ground operation includes the preflight, inspection of the aircraft and the loaded weapons, engine start, extensive system checks, and taxiing in appropriate order to the arming area. The arming area is located near the end of the runway (EOR), where the airplanes will takeoff. Final "go-no-go" checks and the weapons' arming occur at EOR. After the fighters are armed, they are ready for takeoff in the prebriefed sequence.

Relevant Knowledge Sources

The single most important knowledge source is the experienced fighter pilot. The most experienced fighter pilots in the units are designated instructor pilots. Typically, there are four instructor pilots in a squadron of 25 pilots. Experienced pilots, with less experience than instructors, are designated as flight leaders. About one fourth to one half of the squadron pilots are designated as flight leaders. All instructor pilots are automatically flight leaders. Every squadron has a Weapons Officer as its resident expert on weapons and tactics. The Weapons Officer is an instructor pilot who graduated from the demanding four month Fighter Weapons School course at Nellis AFB.

The intelligence personnel maintain classified information concerning enemy capabilities and threat dispositions. They maintain current maps, indicating locations of known threats (such as, SAM, AAA, airfields, troops, convoys), targets and any aerial photographs. The Intelligence

(Intell) Officer uses large maps for the mass briefing. The pilots make copies of the relevant parts of these maps, and use this information in their mission planning. Threat locations and their lethality ranges, depicted as overlays, are some examples of the relevant map data.

The weatherman, whose workstation is physically located in another building, briefs the current and forecast weather conditions for all areas of the mission. He also distributes a hardcopy, commonly referred to as a 'weather flimsy', of the weather data to the pilots. The pilots take this information to their individual planning/briefing room. Weather data critically impacts on route selection (such as, fog in valleys, mountain tops in cloud cover), flight formation (for example, low visibility will cause the flight to fly closer), and type of attacks and weapons delivery events (for example, target area ceiling of 5000 ft. precludes the use of a 30 degree dive bomb weapons delivery event which requires a ceiling of 7000 feet).

Mission Constraints

The tactical mission planning domain is replete with mission constraints, and the process can be viewed as an exercise in constraint satisfaction. A plan which does not violate any constraints, is the desired product of the tactical mission planning effort.

The ATO imposes the initial (high level) constraints. The time on target (TOT) introduces the mission time

requirements. The SCL establishes the aircraft fuel amount and drag index, which directly affects the aircraft fuel flow. Total useable fuel on board the aircraft and fuel flow restrict the fighter range. The term 'Bingo' refers to a specific fuel figure, such as, 1800#, representing the amount of remaining aircraft fuel required to return home with the predefined reserve fuel. Bingo is the point of no return. 'Joker' fuel is a fuel figure, such as, 2300#, representing the fuel amount which warns the pilot of the approaching critical fuel state. This buffer is needed to permit the pilot to plan his departure, or if he is actively engaged with the enemy, to plan his escape.

The development of the overall mission (strike package) plan exposes additional constraints. An OCA mission typically calls for approximately fifty airplanes to bomb an airfield complex, which is three to five miles in diameter. All the TOTs fall within a ten minute time window. This timing requirement forces these airplanes to ingress, attack, and egress the target within close proximity of each other, creating a potentially hazardous situation. To avert possible disaster, the individual flights need to deconflict their respective mission plans. Deconflicting individual plans presents additional constraints (such as, not being in the same piece of sky at the same time). As we travel through a representative flight mission planning session, these and other constraints, will be examined.

Flight Mission Planning

After attending the mass brief (composed of the overall mission, weather, and intell briefs) and receiving the frag, the flight starts preparing their individual plan. The time left for individual flight mission planning is less then one hour; 30 to 45 minutes is typical. Four fighter airplanes comprise the basic strike flight. Appendix C presents a representative division of individual flight member duties. Since newer fighters are single seat (F-16, F-15, A-10), the flight's four pilots perform the tasks previously shared by eight flight members of the two seat fighters (F-4). The F-4's crew consists of a pilot and a Weapon System Officers (WSO).

Target destruction and force survival are the measures of success of any operational mission. To overcome enemy resistance and enhance mission success, certain basic and essential considerations need to be incorporated in every mission plan. The following is a list of these attack mission planning factors:

1. Enemy defenses
2. Terrain
3. Weather
4. Target vulnerability
5. Rules of engagement
6. Force requirements
7. Navigation
8. Formation
9. Munitions
10. Release parameters

A successful mission plan does not incorporate these factors independently, but reflects the interrelationship of these

	DEFENSES	TERRAIN	METEOROLOGY	TARGET VULNERABILITY	ROE	FORCE REQUIREMENTS	NAVIGATION	FORMATION	MUNITIONS	REL PARAMETERS
DEFENSES		X	X	X	X	X	X	X	X	X
TERRAIN			X	X	X	X	X	X	X	X
METEOROLOGY				X	X	X	X	X	X	X
TARGET VULNERABILITY					X	X	X	X	X	X
ROE						X	X	X	X	X
FORCE REQUIREMENTS							X	X	X	X
NAVIGATION								X	X	X
FORMATION									X	X
MUNITIONS										X
REL PARAMETERS										

Figure 2. Mission Planning Factor Matrix

factors. Figure 2, on the preceding page, graphically displays the 45 interrelationships of the ten basic planning factors. The position of each relationship in the matrix does not imply its priority or significance level.

After the flight makes a mission map (cut and paste) and plots the target and enemy defenses, the initial point (IP) is selected. There are many factors, heuristics, involved in the selection of the IP. Target proximity and ease of identification are the major considerations. The IP splits the tactical mission plan into two separate planning tasks, the attack plan from IP to target and the low level navigation route plan, ingress, from the start point to the IP. The flight lead makes a rough estimate of the time and fuel requirements for the attack phase, IP to target, of the plan and presents these values as refined mission constraints imposed at the IP. After the IP is selected, along with the rough estimate of the fuel and time required at the IP, the route is determined by picking the turnpoints. The turnpoint navigation coordinates (latitude, longitude, and magnetic deviation) are extrapolated from the map using a ruler and pencil. Once the coordinates have been collected, they are entered into a hand held programmable calculator or into the squadron Cromemco small computer to compute time, distance, headings, and fuel used for each low level leg. These figures are check and rechecked for accuracy. After ensuring critical mission parameters (constraints) are not

violated, the flight leader integrates the subtotals, such as, individual low level leg times and fuels, ingress route, attack, and egress route totals, into a flight mission plan.

By the time all these distributed duties are performed and the initial plan is made, there is little or no time left for replanning. The remaining time is required to brief the developed plan and get dressed to fly the mission. Since changing one or more turnpoints would require reentering the mission turnpoints, recomputing the values, and redrawing the maps, the present system does not lend itself to easy mission plan modifications. Thus, the present mission planning process is resistant to change and typically allows only one major high level pass on route selection. The required duties of cutting, pasting, and drawing maps, and calculating the mission parameters of times, fuels, distances, and headings, force the pilots to perform mechanical low level tasks. These low level duties, which absorb most of the pilot's mission planning time, detract from gaining mission "situation awareness," which will be extremely important to rely on during the conduct of the mission. This time drain is more acute, since the single seat fighters eliminated the Weapon System Officer (WSO) position in all units, reducing the fighter manpower in half. Mission situation awareness can be enhanced by focusing the pilots' attention on the higher overall mission planning level, where he is constantly working with the overall factors and

assimilating the "big picture". Allowing the pilots to work at the mission level by off-loading the low level tasks, will greatly increase mission awareness, leading to greater mission effectiveness. The pilots would now have time for more direct situation awareness activities, such as, study and 'visualize' the terrain, or address some 'what if' options.

Summary

The tactical mission planning domain is characterized by the following:

1. Time critical environment.
2. Numerous and distributed bits of information.
3. Complex mission planning factors.
4. Complex overall mission integration process.
5. Too much time spent on low level mechanical tasks.
6. Mundane work detracts from mission "situation awareness".
7. An exercise in satisfying constraints.
8. Process resistant to change.

This sample mission scenario was representative of OCA type missions only. Other type missions, some of which rely less heavily on permission planning, were not discussed. However the fighter pilot has to stay proficient in all type missions. The remainder of this thesis will address these shortcomings and propose, implement, and evaluate a prototype tactical mission planning system that exploits artificial intelligence technology to resolve the inherent problems. This prototype will use the takeoff to IP portion of the mission to demonstrate system capabilities.

Chapter three will introduce and describe the current work being conducted in the area of knowledge-based planning systems, focusing on military systems. Chapter four and five will present the conceptual and detailed design of this prototype.

III. Summary of Related Work

Introduction

An engineering oriented definition of Artificial Intelligence (AI) research, elegantly expressed by Jack Mostow, is "figuring out how to bring more kinds of knowledge to bear" on problems that defy traditional computational solutions (Mostow, 1985:1253). Knowledge representation and application, and heuristic search, are the two central techniques that characterize AI problem solving approaches. Artificial Intelligence is subdivided into three main areas; natural language processing, robotics and pattern recognition, including speech and image, and knowledge-based systems, commonly referred to as 'expert systems' (Hayes-Roth, 1984a:13; Rich, 1983:3).

This chapter will address knowledge-based systems, concentrating on military planning systems. The author does not intend to provide a tutorial on artificial intelligence (AI) approaches to reasoning or knowledge representation. The reader is assumed to be familiar with such topics as production systems, both forward and backward reasoning, rules, frames, and constraints. For more detailed information, the interested reader is directed to the bibliography for relevant AI sources, books written by Elaine Rich, Nils Nilsson, Pat Winston, as well as, the three volume "Handbook of Artificial Intelligence", volume one and two by Barr and Feigenbaum and volume three by Cohen and Feigenbaum.

Overview of Planning Systems

Planning can be thought of as a special case of problem solving in which a solution is a sequence of instructions, or operators, for achieving a goal (Stefik, 1980:9), or as "the problem of generating a sequence of actions to accomplish a goal" (Wilkins, 1984:269). A plan is typically composed of subplans; when each is accomplished the main goal is attained. A sequence of instructions or a series of operators is an example of a specific plan, or subplan. An abstract plan is a higher level, general, plan which leaves out specific details. The majority of AI research in planning revolves around autonomous planning systems, which might explain their limited success.

There are no operational military tactical mission planning knowledge-based systems. There are four traditional general planning approaches: nonhierarchical planning (generates only one representation of a plan), hierarchical planning (generates several levels of plans, each successive plan is more detailed than its predecessor), script-based planning (makes use of skeleton plans which are stored, rather than generated), and opportunistic planning (a system that uses a common area called a blackboard, where individual knowledge sources post ideas for constructing a plan) (Cohen and Feigenbaum, 1982:516-519). Chapter XV: Planning and Problem Solving, in volume three of the "Handbook of Artificial Intelligence" by Cohen and Feigenbaum, should be

reviewed for more detailed explanation of these approaches. Initially, each of these approaches has been applied separately in specific domains, primarily academic and research areas. The majority of the recent planning systems are hybrids of the latter three planning systems, hierarchical, script-based, and opportunistic. Research in these areas, and more recently in knowledge-based systems in general, have led to the development of higher-order knowledge engineering (KE) languages, typically referred to as expert system building tools. Some of the generic KE languages available today are: KEE, M.1, S.1, ART, LOOPS. Discussion of higher-order KB languages is deferred until chapter four. Chapter four will also highlight the limitations of these generic tools and contrast the conceptual approaches of these tools with the approach taken in this thesis. The remainder of this chapter summarizes related research. A prototypical planning system that integrates hierarchical planning and constraint satisfying techniques is described, followed by a review of military planning systems.

MOLGEN

Molgen is a knowledge-based system for planning molecular genetic experiments and is the subject of Mark Stefik's dissertation (Stefik, 1980; 1981a; 1981b). Molgen assists in developing gene-cloning experiments, which involve splicing a gene coding, for producing a desired protein, into a bacteria. This is necessary in order for the bacteria to

start producing that protein. Using an object-oriented, frame-based knowledge representation, and a three layered control scheme, Molgen creates an abstract plan and then refines it to a set of specific laboratory steps. It combines knowledge about the user's goal and about genetics, to accomplish this function.

Stefik introduces the problem solving approach termed 'constraint posting.' Constraint posting combines traditional constraint satisfaction concepts with a hierarchical planning problem solving paradigm. A hierarchical planning approach uses a simplified model of the problem. It "suppresses the details of the problem in order to focus on the most important considerations" (Stefik, 1980:9). This last statement is commonly viewed as the 'least-commitment' strategy in hierarchical planning.

Meta-planning, to plan about planning, is Molgen's three layered control structure, which directs the constraint posting process. The three control layers, referred as planning spaces, are termed strategy, design, and laboratory, and are arranged in descending order of control and abstraction. Each space creates and arranges steps in the space below it. The strategy space, the top control layer, coordinates the design steps. This layer implements two different planning approaches; the least-commitment strategy, discussed above, and a heuristic approach that allows the program to make assumptions when a conflict cannot be

resolved. The design space organizes the information about the plausible designer operations. These operators convert abstract objects into specific ones and provides an explicit repertoire of operators for temporally extending plans. The primary objects in this middle level are goal differences and interaction constraints. The bottom space is the laboratory space and contains the knowledge about specific domain objects and operators, such as, laboratory steps, laboratory objects, laboratory operations, and laboratory goals.

Planning proceeded to the laboratory level (deepest level). When constraint satisfaction methods were insufficient to resolve interactions, the problem was elevated to a higher level. The constraints could then be relaxed at the higher level. Acquiring the knowledge about constraints and their interactions is a "bottle-neck" in AI; also noted by Fox in job scheduling problems (Fox, 1983). Hence, autonomous planning systems have yet to be productive. The system design, presented in chapters IV and V, will be similar to Molgen. The most significant distinction is that the end user relaxes the unresolved constraints rather than Molgen's strategy level.

SWIRL

SWIRL (Simulating Warfare In the Ross Language), is a prototype air battle simulation, developed at the Rand Corp. during 1980 and 1982. "The goal of SWIRL is to provide a prototype of a design tool for military strategists in the

domain of air battles" (Klahr et al, 1982: 7). After interactively receiving the required data, offensive and defensive force structures, SWIRL runs the simulation. The simulation represents the conflict created by an offensive force, flying a preplanned route to bomb a target, and a defensive force, attempting to eliminate the penetrators before they reach their target. The user can then observe the air battle, which is graphically displayed on the monitor. SWIRL is written in the ROSS (Rule-Oriented Simulation System) language.

ROSS, an object-oriented programming language, allows the system designer to create the simulation environment using a collection of objects or actors. These objects have slots that contain either specific data or a procedure which determines, and returns, the required data. Exploiting its inheritance hierarchy, Ross facilitates the rapid development and organization of objects by eliminating redundant code. Message passing, a programming technique, describes the communication process between actors. SWIRL's simulation is based on this message passing capability. "Message passing provides the basis for understanding complex interactions between objects" (McArthur and Klahr, 1982:1). The system designer, using Ross, can define the behavior of the relevant objects or actors, and place that behavior, a procedure or set of procedures, inside the appropriate slot. This modular programming style helps the user run the

simulation and modify inappropriate behaviors.

TATR

TATR (Tactical Air Targeting Recommender) is a prototype knowledge-based system, developed at the Rand Corp., during 1979 through 1984, inclusive. The primary functions of TATR, an interactive program to be used at the USAF Tactical Air Control Center (TACC), are "to provide a plan for attacking enemy airfields and to project the effects of implementing the plan" (Callero et al, 1984:v). This system can project the effects of implementing a particular plan over a period of time, typically a few days. The results of this projection are used to revise the original plan, if necessary. TATR is written in the ROSIE (Rule-Oriented System for Implementing Expertise) language. Rosie, like ROSS, uses an English-like syntax; a desirable feature which helps non-programmers understand the heuristic logic associated with TATR or any other similar system. TATR is a rule-based, forward chaining system. The rules were developed, based on information provided by experienced air targeteers (Callero et al, 1981:3). They represent the domain experts' heuristics, rules of thumb.

Using multiple menus, TATR, initially, assists Air Force tactical air targeteers select and prioritize targets. After completing this "Target File Generation" phase, specific target elements are determined. The "Targeting" phase, also identifies the desired effects and best friendly

resources to achieve these effects. The next phase, "Force application consists of producing a plan matching friendly air resources and enemy target elements" (Callero et al, 1984:4). The preparation of the Air Tasking Order (ATO) is the final phase. The transmission of the ATO starts the fighter unit mission planning process, described in chapter two.

KNOBS

KNOBS (KNOledge-Based System) is a backchaining production system, integrating rule and frame inference architectures, with an English language interface (Engleman et al, 1979:247; Engleman et al, 1980:184). This system was constructed "in support of planning ground-strike counter-air missions, in which aircraft are sent to damage targets such as enemy surface-to-air missiles, airfield runways, fuel dumps, etc." (Engleman et al, 1983:450). KNOBS, essentially, performs the same tasks as TATR's middle two steps, targeting and force application. KNOBS integrates knowledge about targets, resources, and missions, in developing mission plans and then checks those plans for consistency. It also helps rank alternative plans or generate new plans.

KBS

KBS (Knowledge Based System) was developed by Mitre Corp. to provide a demonstration prototype which would assist a staff officer develop plans in response to crisis

situations (Benoit et al, 1982:3). "An Air Show of Force" was the sole crisis scenario developed, and only a minimal graphics user interface was implemented. Although KBS is an autonomous system, it can be used in an interactive mode. KBS employs the AI techniques of hierarchical planning, frame knowledge structures, and a rule base. The majority of the knowledge, that is, rules, was "obtained from experienced military strategists through interviews and direct observation of their methods" (Benoit et al, 1982:1).

Summary

The tremendous gains in basic computer technology, in the last five years, have made commercial applications of AI technology feasible, as well as profitable. The application of knowledge-based systems to solve traditionally complex, 'hard', problems is slowly becoming a reality. The main components of an artificial intelligence approach to problem solving is knowledge and heuristic search. Autonomous operation, with provisions for interactive use, appears to be the basic design concept of the current planning systems. Interviewing and observing the domain expert, has been and still is, the typical way of acquiring the knowledge necessary to design any system. Current knowledge-based systems are hybrids of the different AI paradigms (rule-based, frame-based, or object-oriented). All these systems were, and are, just isolated research products. Each system is limited in its scope of design, taking a microcosmic rather

than a macrocosmic system view, in the analysis of the problem domain.

The results of such an approach are systems that never make the transition from the research labs to the field, the end users. The only benefits received from paying the high cost of using these systems is the shallow end product; for example, a list of targets, a list of airplanes with their respective weapons' load, a specific plan. The previously cited planning systems were research tools. The researchers wisely chose domains of interest to the research funders. This often has the unfortunate effect of prematurely raising expectations about near term solutions, using artificial intelligence techniques. The simple fact is autonomous planning systems are laboratory curiosities. They are not ready to be applied to real world problems. Earl Sacerdoti, in a recent article, highlights the important issue of technology transfer (Sacerdoti, 1985b).

Not only are autonomous planning systems unrealizable, an in-depth analysis of the problem domain (chapter two), current tactical mission planning at the unit level, has identified the main problem areas that need to be resolved. This chapter reviewed and analyzed recent work related to this thesis. The next chapter presents the conceptual design of an interactive planning system that exploits domain knowledge and whose benefits are farther reaching than a single shallow product; a mission plan. Chapter five will

transform the conceptual design into a detailed design of a tactical mission planning prototype, ready to be used in tactical fighter squadrons; this supports the engineering technology transfer concept.

IV. Conceptual Design

Introduction

The primary goal of tactical aviation is mission accomplishment with force survival. The design of any system in this domain has to directly support this objective. A macrocosmic analysis of the tactical fighter pilots' domain has to include the actual flight of the planned mission. Given the complexity, and the limited time available, it is not sufficient for designed systems to minimize planning time; they must increase the utility of the time used. This adheres to the 'get more for less' Air Force principle. This last point is crucial for the technological transfer of the system to the operational environment. More important than mere technological transfer is the issue of technological acceptance. A delivery system will not be successful if it is not accepted and used. Acceptance is contingent upon the system's operational utility. This is a basic software engineering issue addressed later in this chapter.

The single most important concept to comprehend for successful system design and development in the tactical fighter domain is "Situation Awareness." **Situation awareness is the domain master all must serve.** An activity that does not support and/or increase mission situation awareness is not cost-effective and will not be accepted or utilized. The next section on situation awareness and human information processing is essential reading if one is to get a

reasonable understanding of the fighter pilot's world. This thesis proposed, implemented, and evaluated a working prototype tactical mission planning system. This prototype not only produces a more refined mission plan than generated by current operational planning systems; its use in the squadron should directly increase mission situation awareness, essential for the flying portion of the mission.

Situation Awareness and Human Information Processing

Situation awareness is the paramount ingredient for successful task accomplishment in any domain; "however, it's a difficult subject to address because of its nebulous nature" (Waddell, 1979:3). Since 'situation awareness' is a recurring theme in the fighter pilot domain, an in-depth analysis of this phrase is warranted. Situation is a combination of circumstances at a given moment. Awareness is having knowledge or cognizance; implying knowing something either by perception or by means of information. Perception is the process of achieving understanding, directly through any of the senses, especially sight or hearing. These basic definitions, which were extracted from the American Heritage Dictionary, start our analysis.

Situation awareness is the dynamic mental model of the world, an individual's frame of reference used to keep himself oriented. The 'situation' represents a single time slice of that dynamic environment. This mental model of the world may be thought of as our link with reality. Human

performance is directly related to an individual's situation awareness level. Everyday judgments and decisions, while relating to the real world, are based on information contained in our mental model. The greater amount of world information (knowledge) we have, the more complete our model. The more complete our model, the more accurate and appropriate are our decisions. Knowledge represents the single most important element directly impacting the completeness of the perceptual world model. The situation awareness level is directly proportional to the amount of world knowledge contained in our model. Understanding how man acquires knowledge and processes information is therefore essential to fully comprehend the meaning of situation awareness.

There is no theory or model that adequately explains the complexity and flexibility of the human thought process; therefore, a descriptive approach will be used to operationally discuss human information processing (Santilli, 1985:6). Lt Col Santilli, currently chief of the Human Factors Mishap Analysis, Function Crew Technology Division, USAF School of Aerospace Medicine, Brooks AFB, Texas, decomposes the information processing task into four steps; sensation, perception, decision, and response.

The sensation step entails our sensory collection of raw data. In the flying domain, sight and hearing are the most important senses. Raw data, input at the senses, will

decay very rapidly (as fast as .5 seconds) unless it is preserved by means of conscious attention (Santilli, 1985:7). Since there is more sensory data than the average human can process, the inputs must be prioritized, and only those deemed significant will be attended to, or processed.

The actual processing step takes place in short-term memory (STM); where information remains for approximately fifteen seconds before it too will decay. By contrast, long-term memory (LTM) is a more permanent storage location. The information stored there is generally referred to as our knowledge base. However, for information to reach LTM, "requires rehearsal, training, and integration with previous knowledge" (Santilli, 1985:7). Memorizing a sequence of procedures, 'Critical Action Procedures' (CAPS), is an example of transferring information to LTM. CAPS are used in Tactical Air Command as initial corrective actions to many fighter emergencies. These actions may be done either on the ground or in the air, after the pilot properly identifies and analyzes the situation.

Attaching meaning, semantics, to new information that has been attended to characterizes the perception step. This step involves a series of conscious level processes. First, identify the specific piece of information; secondly, assess its' relationship with other information in both STM and LTM, and finally, factor in the current world state, internal (psychological) and external. Thus, the perception

step transforms information into knowledge; with a 'chunk' of stored knowledge, the resultant product. These chunks are templates which represent a specific situation, a time slice of the actual world environment. Each template, which encapsulates a significant experience, is later accessed when processing new information. During the sensation step, new information is pattern-matched against our templates, so if a match is found, further time consuming, conscious level processing can be avoided. The main difference between an instructor pilot, the domain expert, and a non-instructor pilot is the IP has more "chunks", templates of knowledge, which represent his experience. The importance of these templates, used in the pattern matching process, will be highlighted in the discussion of the decision-making process, especially under stress.

Having perceived a situation, the decision step determines whether further actions are required and, if they are, which actions are appropriate. This analysis draws heavily on past experience and training. It integrates previous knowledge with new information about the current world state, and produces a decision. If past experience and training is limited, or old, conscious mental activity is required to process the information, which represents the current situation, before a decision can be reached. Searching the limited knowledge base for similarities, pattern matching, and updating, and refreshing old knowledge

are processing examples. This active mental process is time consuming, resulting in a delayed time critical decision. By contrast, when the current, perceived, situation is similar to past experiences, that is, a familiar environment, the processing time is minimized and a quicker decision is reached.

The response, the actual physical action, is the last step of the process. An important term associated with the response is 'reaction time.' Reaction time is typically defined as the amount of time from the perception step to the response step. To complete our understanding of situation awareness, an analysis of human behavior or performance, while formulating a response, is necessary.

Humans are goal-oriented creatures, who actively select their goals, and then seek the appropriate information required to accomplish those goals (Rasmussen, 1983:257). Human behavior in a familiar environment will be oriented toward the goals and controlled by a set of rules. This set of rules, similar to scripts or templates, is predefined, typically memorized or habituated, courses of action that can be implemented without inductive or deductive reasoning. Rasmussen's use of rules should not be confused with rule-based reasoning, an AI technique which could be used to implement human-like problem solving at this or higher levels. However, in unfamiliar situations, where proven rules are unavailable, the behavior becomes goal-controlled.

That is, the current information is integrated with previous knowledge, generating a proper sequence of actions to accomplish the goal. If this new sequence is successful, it will be used to update one's existing set of rules. Thus, "the efficiency of humans coping with complexity is largely due to the availability of a large repertoire of different mental representations of the environment from which rules to control behavior can be generated ad hoc" (Rasmussen, 1983:258). Rasmussen categorizes human behavior into three typical levels of performance: skill-based, rule-based, and knowledge-based.

Skill-based behavior, basic sensory-motor activity, requires no conscious control. While performing a skill-based task, the senses are automatically focused on specific pieces of information about the environment needed to subconsciously update our mental world model, and properly orient ourselves with the real world. The cost of processing this type of information and performing skill-based level activities is quite low with regard to the amount of human conscious processing time. Human activity can generally be considered a sequence of skill-based tasks; subroutines, integrated to attain a specific goal or goals. Walking or running are common examples of such behavior.

The next level of behavior required for the actual composition phase, developing an appropriate sequence of tasks to attain a given goal, depends on the individual's

familiarity with the task or situation. Behavior, in a familiar environment where known skill-based subroutines are readily available, is controlled by a 'stored rule' or procedure. This rule may have several sources. It could have been derived empirically during a previous similar situation, or the rule itself could have been communicated by a more knowledgeable source. An instructor pilot, or a pilot with a few years of experience, is such a source. He conveys his proven rules to the inexperienced, new pilot. The rules and procedures written in the flight manuals and regulations are other sources. Thus, rule-based performance is typically based on explicit know-how and is a relatively inexpensive information processing activity.

However, when faced with an unfamiliar situation, where there are no a priori set of rules, performance must move to the most 'expensive' conceptual level, the goal-controlled, knowledge-based level. During knowledge-based performance, the goal is explicitly formulated, a plan is developed, and the effects of the proposed plan are tested with respect to goal attainment. This entire process is expensive because it is time consuming. You can compare the cost of processing information in a new, or unfamiliar, situation as the initial 'start-up' cost for any system.

The final two factors that directly affect information processing are an individual's level of awareness and his level of attention. The level of awareness, the cognitive

level at which mental activity takes place, is further subdivided into the conscious, preconscious, and subconscious levels. Only the first two levels will be discussed. The conscious level is where active thought processing takes place, for example, reasoning and decision-making. The preconscious level, the more passive activity level, is the repository of the STM, LTM, and established habit patterns. Preconscious patterns, which are stored in LTM, require repetition. "Any activity that requires active information processing or decision-making, must take place at the conscious level of awareness" (Santilli, 1983:8).

Level of attention, the degree to which the conscious level of awareness is being used, can be described by three terms; span of attention, focus of attention, and margin of attention. Span of attention is an individual's total capacity to handle information at the conscious level. Human information processing 'bandwidth' is a common synonym for span of attention. It comprises both how much information can be processed and for how long. The focus of attention is that portion of a person's span of attention being used at any given time. Activities require different levels of attention and therefore, have different conscious processing costs associated with them. The margin of attention is the value representing the difference between the span of attention an individual possesses at the time, and the focus of attention required to perform the required task. When the

margin of attention goes negative (that is, the bandwidth is insufficient to completely process the required information), then the individual is cognitively task saturated. When a pilot is task saturated he cannot function properly in the aircraft. Typically, task saturation is manifested as deteriorated pilot performance; he will start to commit errors of commission or omission. Task saturation has become a serious contributing factor in many fatal aircraft accidents.

In summary, situation awareness is the dynamic mental state which contains sufficient knowledge to completely represent the world. This knowledge keeps one properly oriented with one's environment. Situation awareness is the most important attribute a fighter pilot can possess; it directly contributes to successful mission accomplishment, especially, in the high speed, low altitude, and threat-ridden flying domain. Maintaining good situation awareness requires continuous updating of the mental world model, that is, processing the appropriate information. Relevant knowledge, quickly accessed with little or no reasoning, is the core ingredient of situation awareness. The human processing bandwidth has a finite limit. When it is exceeded, task saturation occurs, resulting in increased pilot errors. Human information processing and performance is accomplished at different levels. Each level has an accompanying focus of attention cost. Skill-based behavior incurs the least cost,

and knowledge-based behavior incurs the most. Familiarity with the current situation, expressed by similar previous experience, training, or knowledge, reduces the information processing time and elicits the relative low cost rule-based behavior.

A typical tactical fighter mission, flown low to the ground, at speeds exceeding 600 mph, is a very demanding task. The terrain, a major part of the environment, is constantly changing at a very rapid rate. Maintaining situation awareness, commonly referred to as "staying ahead of the aircraft," requires a major portion of one's span of attention in order to keep from impacting the ground. This reduces margin of attention, dramatically shrinking the remaining bandwidth. The error margin due to inattention or distraction is extremely small (The time to ground impact from 100 feet above ground level 'AGL', flying at 500 knots, with a one degree descent angle, is 7.2 secs; with a four degree descent angle ground impact will occur in 1.8 secs). Obviously, staying oriented with the environment is imperative for survival. Mission factors and aircraft parameters are always changing; fighters fly the terrain not straight lines on a map. The aircraft 'G' loading, position, and flight vector are continuously changing. This information also needs to be processed and their effects on the mission goals assessed. This information processing activity further reduces the margin of attention. Having analyzed the

relationship between focus of attention (FOA) and span of attention (SOA), task saturation can be mathematically represented by the following inequality:

$$\left(\sum_{i=1}^{n} \text{FOAi}\right) > \text{SOA}$$

where n is the number of tasks.

The strategy required to effectively cope and survive in this unforgiving, low error-tolerant environment, is to increase the margin of attention and reduce the focus of attention required to process additional information. This increased margin permits the pilot to safely deal with contingencies. Aircraft systems malfunctions, weather changes, and unbriefed threats are representative factors demanding plan modification. Minimizing knowledge-based behavior directly supports this strategy. Increased experience, training, knowledge, and thorough mission preparation increases situation awareness, and directly supports rule-based processing behavior by familiarizing the pilot with the mission environment. Prior to designing a ground based mission planning system, which directly supports an increase of in-flight situation awareness, some basic system requirement issues have to be resolved. Should the mission planning system be autonomous, with interactive capabilities, or be designed as an interactive aid for tactical mission planning, off-loading the low-level, disjoint, isolated,

laborious chores, highlighted in chapter two, onto the computer? This dilemma is the subject of the next section.

Interactive Versus Autonomous System

Active pilot involvement in tactical mission planning is essential for successful mission accomplishment. Squadron, ground, mission planning systems must be designed to utilize the limited, alloted mission planning time in the most cost-effective manner. The planning process has to use the pilot's time wisely. Focusing, or channeling, the pilot's attention and activities on the high level, mission oriented planning aspects, facilitates the increased assimilation of mission essential information. This knowledge assimilation represents information transfer to long term memory and develops some predefined contingency rules, on which the pilot bases many critical in-flight decisions. The maximum use of these types of activities builds a pilot's experience base, increasing his situation awareness, contributing to accurate, appropriate, and timely decisions. Therefore, it is not sufficient for mission planning systems to be interactive, but, more importantly, these systems have to be designed as interactive. Autonomous mission planning systems deny the pilot valuable mission essential knowledge, on which he would base in-flight decisions. This knowledge deprivation, decreased situation awareness, may well cost the pilot his life.

A Knowledge Based Mission Planning System

A knowledge-based system is a computer program that contains large amounts of specific knowledge about a particular problem, or domain. It uses heuristics, domain rules-of-thumb, "to focus on the key aspects of particular problems and to manipulate symbolic descriptions in order to reason about knowledge they are given" (Harmon and King, 1985:7). Knowledge is at the heart of the system. There are essentially two types of knowledge, public and private. Public knowledge, as its name indicates, is the collection of published information; facts, theories, and basic definitions, which represents part of a domain and is universally available. Private knowledge, by contrast, is the unpublished part of human expertise. It is comprised of human judgement, and personal, proven rules-of-thumb, that are applied to resolve an, otherwise, intractable problem.

It has been shown in planning domains, such as, described in this thesis, a combinatorial explosion occurs as the number of constraint overlappings, caused by the interactions of mission factors, increases (Norman, 1985). "The growth rate of an exponential function is so explosive we say a problem is intractable if all algorithms to solve that problem are of at least exponential time complexity" (Aho et al, 1974:364). Finding the optimum path from a fighter's home base to the assigned target, including the optimum flight parameters of airspeed, altitude, fuel flow, and

threat avoidance, is a set covering problem. This set represents the optimum combination of all mission factors, that is, the set covers all combinations. A set covering problem is known to be NP-complete, 'Nondeterministic polynomial-time complete' (Aho et al, 1974:378). When faced with a large number of possible solutions, that is a large search space, skilled application of heuristic knowledge eliminates unpromising areas from consideration, pruning the search space and producing an appropriate solution. Symbolically encoding heuristic knowledge and applying this knowledge to limit the search space, are the main features that differentiate a knowledge-based (AI) approach to problem solving from conventional programming approaches. Thus, "knowledge about the domain is the key to more efficient solution methods, developed for delaying and moderating the inevitable combinatorial explosion" (Nilsson, 1980:7).

Domain knowledge "consists of (1) the symbolic description that characterize the definitional and empirical relationships in a domain and (2) the procedures for manipulating these descriptions" (Hayes-Roth et al, 1983:13). To be of value, this knowledge has to be structured in a program. The analysis of the problem domain, presented in chapter two, suggests the domain knowledge base can most efficiently be represented as specific objects or actors. Furthermore, these objects can be organized hierarchically, effectively applying the principle of inheritance, that is, sharing

common properties, knowledge, and procedures. Mission objects such as, fighters, sams, nav-legs, turnpoints, runways, tanks, weapon delivery events, mission maps, fuel monitors, time monitors, and system monitors, are represented as frames. The knowledge, embedded in these objects, is represented in several different ways. Slots contain knowledge represented as static values, procedures, and rules fused with procedures. A runway-target object has a slot containing navigational coordinates, identifying its location. A target knows the type of weapons that can destroy it. A fighter has a slot containing a procedure to compute fuel flow. A fighter-base has a procedure, containing rules, to select the appropriate runway to use, depending on the local winds input by the weather report. Thus, a hybrid knowledge representation scheme within an object-oriented paradigm is most appropriate. Other benefits of object-oriented programming techniques will be presented in the software engineering section of this chapter.

The primary goal of applying a knowledge-based approach to tactical mission planning is to embed as much domain knowledge into as many objects as possible, producing an intelligent, 'smart', planning environment. The following example illustrates the difference between current 'dumb' environmental objects and 'smart' objects. The goal of this exercise is to get the navigational coordinates, latitude and longitude, of a proposed turnpoint in a format readily

usable in the domain aircraft, in this example, an F-16. The pilot gets a standard map, pencil, and ruler. He selects the turnpoint, draws two lines, one from the point through the latitude indices and the other through the longitude indices, drawing a large '+', with the turnpoint at the intersection. The pilot interpolates the tick marks, drawn at each index, and converts the coordinates to the acceptable format. This manual process, typically, takes more than sixty seconds. Using a 'smart' map, the pilot just points to the location he wants, using a mouse, "an electronic grease pencil," or other common computer hardware device, and 'shoots', that is, pressing the appropriate mouse button or other such facsimile. The navigational coordinates are displayed, in the appropriate format, in less than one second, a substantial saving in planning time. Along with these coordinates, the point's elevation is displayed. More detailed examples of the intelligent capabilities of this system will be presented in the next chapter. Since a human can concurrently maintain only four to seven chunks of information in short term memory, this massive system distribution of knowledge aids the pilot in taming the complexity of this domain. An example of this last point will be presented in the constraint monitoring section of this chapter.

Hierarchical Planning Approach

Abstraction allows one to focus on the most important,

high level, considerations of a particular problem. The navigational planning problem introduced in the previous section will serve as the vehicle for highlighting the important relationship of hierarchical decomposition and problem abstraction in solving, otherwise, intractable problems.

Decomposing a plan into multiple layers of subplans, each layer adding more detail, characterizes a hierarchical planning approach. At the top layer, most abstract, only the most important factors are considered. The pilot gets the mission map and measures the straight-line distance, between home and target; he computes the approximate fuel and time costs; and very quickly determines if this initial plan is feasible. If the plan is acceptable, the process proceeds to the next level, abstract layer, of detail, refining the previous plan. The pilot decomposes the flight into sectional legs, selecting specific areas to fly. The plan is again checked for feasibility and consistency. If a plan, or subplan, is not feasible, it can be immediately abandoned or modified before too much effort has been expended. The designed interactive hierarchical approach, paralleling the pilot's approach to mission planning, produces an acceptable plan more efficiently than the current disjointed, computer-aided, manual process.

Constraint Posting and Monitoring

Chapter two also elucidates the constraint driven

nature of the tactical mission domain. The ATO immediately posts the most demanding mission constraints. It establishes a 'hard' TOT, time on target, and the aircraft configuration, from which the aircraft's usable fuel and drag index are computed. The term hard is used to express the inflexible nature of the assigned time when the target is to be attacked. The number and type weapons to be carried, included in the weapons load (SCL) portion of the ATO, determines the maximum carriage and employment aircraft airspeeds. The TOT is the single most demanding constraint in tactical warfare. The target's location, also included in the ATO, is used to compute the total mission distance. The known enemy defenses and early warning radar locations, terrain, and weather are other mission factors imposing constraints on the mission plan. The above examples represent the major, though not all, factors and their associated constraints that impact the initial abstract plan.

It is apparent from the number of factors, their interrelationships (graphically depicted in figure 2), and the associated mission constraints, that the planner's short term memory capacity is easily exceeded. Embedding the knowledge represented as procedures, rules, and constraint values into the objects which comprise the domain environment, transfers the task of constraint monitoring to the system. A computer is better suited to keep track of these details. Off-loading this drudgery, resolves the costly

problem portrayed when the specific piece of knowledge, required to evaluate the interaction of constraints, is not in the pilot's short term memory, forcing the pilot to access other sources and replace parts of STM. This costly process of replacing the information in STM is analogous to the "page swapping" problem associated with operating systems having a "demand-paged memory management" scheme (Madnick and Donovan, 1974:139-144).

Pilot Relaxes Constraints

The pilot can now spend more time on mission essential activities, directly contributing to situation awareness. He can fully focus his attention on mission level objects, such as, the mission map or the target area. This focused attention lends itself to a greater amount of information being transferred to long term memory. The pilot gets alerted only when conflicts cannot be resolved, a role which is analogous to Molgen's strategy level (Stefik, 1981b:156-160). Where the strategy level control structure did not have the power to resolve high level constraint violations, the pilot, after being appraised of the situation, can relax those constraints. The following typical scenario will clarify this last statement.

Assume a detailed mission plan has been created that meets all constraints except one; the mission reserve fuel will fall 500 pounds below the required amount. This plan is unacceptable in its present form and cannot be modified

to conform to the prescribed constraints. However, the pilot knows if he uses ten seconds less afterburner on the attack and ten seconds less afterburner on the escape maneuver, he can save 600 pounds of fuel. Another option is for the pilot to modulate his use of the afterburner, rather than selecting full afterburner for most of his maneuvers. This alternative will also save the pilot approximately 600 pounds of fuel. Thus, due the pilot's intervention, the plan was completed, accepted, and successfully flown. However, there is always a chance other constraints will be violated; for example, using less afterburner decreases airspeed, which, in turn, causes the pilot to be in the target area longer, increasing his exposure time. But this is exactly the role of 'what-if' capabilities designed into the system. The pilot can consider these options and formulate appropriate rules.

Software Engineering Principles Paramount

The success of any programming project, conventional or AI, can be traced to the disciplined application of basic software engineering principles. Although, this section is not meant as a tutorial on software engineering, some of the relevant basic principles will be discussed. The requirements definition phase is the first step in successful system development. A complete, consistent, and unambiguous product specification is the primary goal of the requirements phase. This functional description outlines the

required software functions, interfaces, and performance. It specifies what the software product will do, not how it will do it. The requirements definition serves the primary needs of three groups of people; the system designer, interested in understanding the structure of the problem so as to better create the required software product, the customer, actually paying for the product and not necessarily using it; and the end user, concerned only with the usability of the product and not necessarily the buying of it. The designer, buyer, and user, form the 'dreaded software engineering triangle', a term the author coined to describe the phenomenon of 'things getting lost in the middle.'

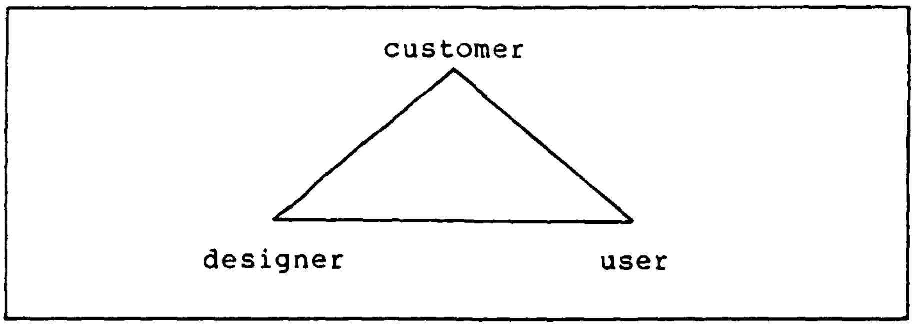


Figure 3. The Software Engineering Triangle

The importance of domain expert and end user's active involvement in the requirements and design phase of software system development cannot be overstated. Earl Sacerdoti in his presentation as a keynote speaker stated, to have a successful expert system, the domain expert has to be actively involved fifty percent of his time during the requirements and design phase and then twenty-five percent

thereafter (Sacerdoti, 1985a). Tom Garvey, SRI, and Ed Taylor, TRW, also keynote speakers at the same conference, expressed similar views. The author feels these percentages should be viewed as the bare minimum. Without this type involvement a project is destined to suffer the fate of other poorly designed and engineered products, disuse.

The danger of the software engineering triangle dilemma will be highlighted by the following brief example. The product is a knowledge-based system to aid the fighter pilot in the single seat tactical fighter domain. The domain experts, also end users, are the fighter pilots. The customer is the contract monitoring agency. The system designers are the aerospace companies.

The designers are working for the customers and will do their best to meet their specifications and needs. However, the customer, since they are not the actual end users, cannot realistically determine user requirements. If the end users are not actively involved in the requirements and design phase, the designers will receive the customer's, typically management, perceived specifications. Industry, with the contract monitor's guidance and minimal interfacing with the fighter community, will develop AI systems, perceived to solve the operational fighter needs. Fighter pilots, the end users, will be expected to accept and fly with systems, they have not helped design.

The logical conclusion of this scenario indicates all

three relationships will be strained and each member of the triangle will loose the respect and confidence of the other two. This scenario can be avoided if the requirements phase adheres to the basic software engineering principle prescribed in this section.

The requirements process leads to the subject of requirements definition specifications. Beside the standard specifications' environment; informal, formatted, and formal, Shiel's introduces a fourth specifications' environment, "exploratory programming" (Shiel, 1984:19; Mostow, 1985:1253). When the domain is characterized with poorly stated, and frequently changing requirements specification, as is the case in the AI application domain, a quick prototyping capability is a must. Only after observing and analyzing the functioning interactive prototype, can the true system requirements be specified.

Object-oriented programming techniques produce programs that consist of a set of objects that interact with one another by sending messages. Frame-based systems are inherently object-oriented, since each object can effectively be represented as a frame. As discussed earlier, the frames contain slots, into which knowledge needed to responded to a message is stored. Message passing capabilities help the system designer and user to better understand the complex interactions between system objects, also increasing system situation awareness. Storing behavioral responses in these

attribute slots facilitates modular programming, a basic software engineering principle.

A Modular programming style supports the most important requirement for any program unit, that is, a particular function or module, to address only a single, logically coherent task. The modular knowledge structure of objects also supports the decomposition criterion known as information hiding. Each object, a system module, is encoded with the knowledge to function appropriately. The following example will clarify these ideas. Some of the objects, such as, a runway, turnpoint, or sam, need to be drawn on the mission planning map as class unique images. Since the information is embedded in the objects the user does not need to know how to draw a desired object. He simply sends the object a message, containing the location, to draw itself. The object contains all the required information to perform the requested task. Another example of the above two principles addresses object behavior. Subclasses of aircrafts each have their own aircraft specific procedures for computing weight, speed, fuel, and drag index. An F-16 computes mission parameters differently than a KC-135, tanker. The program simply sends any aircraft the exact same message, not worrying about what type of aircraft it is. The responsibility and knowledge of how to decode that message and return the appropriate information is contained in the object. The obvious benefit of keeping objects and

modules logically coherent is realized during program modification.

Screen information management, is the last software engineering related topic. The user-interface dilemma of what type, and how much, information to present the user is an important requirement and design issue. When the competition for screen, that is, the CRT monitor, space increases, multiple screens is a viable solution. The next chapter will highlight the requirement of two screens, needed to unclutter the mission planning environment and effectively interact with the pilot.

<u>Higher Order Knowledge Engineering Language(s) Required</u>

Artificial intelligence application systems' development occur in the realm of 'exploratory programming.' The development of a tactical mission planning system falls within this environment. In a recent article, Jon Doyle contrasts AI and traditional approaches to application development, summarizing in favor of AI: "the AI approach permits relatively rapid and cheap development of a prototype" (Doyle, 1985:1389). Rapid prototyping is a major consideration in identifying and selecting an environment conducive to exploratory programming, knowledge engineering. A review of the environmental requirements for mission planning will guide the selection and/or development of a higher-order knowledge engineering language, a form of knowledge engineering building tool, to develop and build the prototype.

The tactical mission planning system has been characterized as hybrid knowledge-based system. It includes the use of frames, rules, and procedures to represent domain knowledge. Objects, containing this knowledge, define the system database, highlighting the need for object-oriented programming. The need for symbolically processing knowledge suggests the use of Lisp as the basic programming language. The dynamic nature of the development environment, characterized by the rapidly changing specifications, demands a higher-order language to relieve the system developer, and later the prototype user, the complex and time consuming tasks of creating, modifying, and observing system activity; essentially increasing their system situation awareness.

The Symbolics 3600 series system was the natural obvious choice for the basic system, hardware and resident software, to design the prototype. The major selection feature of the Symbolics' lisp environment includes the programming language, ZetaLisp. The powerful coding development tools offered by the Zmacs editor is a tremendous time saver. The other time saving feature is the partial function or list evaluator capability resident in the editor, supporting incremental program development. The flavors package offered a frame-based structure capability. The bitmap editor supported the creation of graphic icons representing tactical mission objects.

KEETM (Knowledge Engineering Environment, IntelliCorp)

was selected as the 'expert system building tool', a higher-order language which functions in concert with the Symbolics' environment. KEE permits the rapid development of a domain knowledge base. Using KEE as the vehicle to present their views, Fikes and Kehler, in a recent article, state the basic criteria for a knowledge representation language are; expressive power, understandability, and accessibility (Fikes and Kehler, 1985:904). This article is suggested as an introductory tutorial in frame-based knowledge representation scheme.

Although KEE, initially, appearing to be promising, release 2.1 was not the panacea to the important user-interface aspect of the mission planner. The remainder of this paragraph will briefly describe the limitations of KEE as applied to this domain. It is assumed the reader is familiar with the Symbolics-Lisp environment, and the concept of windows and their processes. All KEE activeimages, including icon images, are implemented as lisp windows. This presents several disadvantages. When one window overlaps, even another pixel, the window on the bottom is 'de-exposed' and its process is deactivated. The mission contour map, to be fully described in the next chapter, is a window with an associated process that makes it a 'smart' map. The icon images, which represent domain objects, (for example, turnpoints, tanks, targets, airfields, etc.) are placed on the map, de-exposing the map and simultaneously,

deactivating the high-level planning capabilities. A second important disadvantage of the window implementation of icon images is the high cost of processing time incurred anytime the image is moved. Moving a turnpoint, and refreshing or redrawing the screen are examples of this cost. Presently, KEE, release 2.1, can only support one monitor, denying the use of multiple monitors to unclutter the main screen, which is another major limitation. It is the author's understanding that these limitations have already been identified and resolved in release 3.0, due late 1985 or early 1986.

The solution for the 'higher-order knowledge engineering language' requirement was to develop a custom hybrid language for tactical mission planning, combining the Symbolics' environment offering Zetalisp, flavors, graphics, and Zmacs editor, with KEE the knowledge engineering development tool, offering rapid prototyping. Several modification had to be made to KEE, permitting the use of its objects on multiple monitors. The second monitor used was the Symbolics high resolution color monitor. The author successfully created and used KEE activeimages simultaneously on both the regular 3600 system monitor and the color monitor, including the manipulation of the digiactuator panels, displayed on the color monitor. All icon images were implemented using the bitmap editor, the flavors package, and the 'bitblt' resident functions of the graphics package. Bitblt'ing, using the XOR option, efficiently and quickly draws

and erases screen images.

Summary

Situation awareness is the paramount ingredient for successful tactical mission accomplishment. Supporting or increasing situation awareness is a major objective or requirement for any system designed in the tactical aviation domain. Interactive ground mission planning systems can synergistically utilize the fighter pilot's limited planning time. A more refined mission plan and increased situation awareness are the most beneficial products.

A hybrid mission planning system, including frames, rules, and procedures, is proposed. An object-oriented paradigm will be used exploiting basic software engineering principles of modular programming design and information hiding. Knowledge, represented as static values, rules, and procedures, will be embedded in mission objects. A hierarchical approach to problem solving, with the pilot relaxing constraints, will aid the pilot to cope with the dynamic and complex environment. The Symbolics 3600 series system, with a high resolution color monitor, will provide the prototyping environment. An author modified version of KEE, release 2.1, will be used as a system building tool.

This prototype will be instrumental in generating accurate delivery system requirements, supporting industry's system requirements definition phase. Once these requirements are elucidated, the delivery system can be developed

using any appropriate hardware or any standard programming language.

The next chapter will describe a working tactical mission planning prototype, incorporating the conceptual design issues presented in this chapter.

V. Detailed Design

Introduction

Chapter four presented the conceptual design of an interactive mission planning system that increases pilot situation awareness, while producing a viable tactical mission plan for an Offensive Counter Air (OCA) mission. This chapter will describe the working prototype mission planning system. The prototype was developed on a standard delivery Symbolics 3670 system with a Symbolics, high resolution, color monitor, using the eight bit color software package. KEE, release 2.1, was used as the basic knowledge engineering language. Modifications to KEE had to be made to create and use KEE ActiveImage panels on the Symbolics color monitor. The prototype has been demonstrated on both the Symbolics 3670 and 3600 basic systems. Each Symbolics' system can have a different size monitor, that is, different dimension in pixels. The prototype has been designed to be monitor size (hardware) independent. Functions that build windows, such as, those containing mission maps, first query the hardware to determine screen size, allowing the software to be portable.

The next two sections will present the actual prototype planning environment. They will highlight the reasons for some design and implementation decisions. The remainder of the chapter, starting with the 'Mission Maps: The Planning Environment' section, will step through a typical mission

plan. It will cover relevant aspects of a flight from takeoff to the IP. An example of a 'last minute' replanning scenario demonstrating the capability to react to a newly discovered SAM, will be presented. This chapter is not meant to be a tutorial on Lisp programming or the use of KEE. The interested reader is referred to lisp programming texts by Winston, Wilensky, and Charniak (Winston and Horn, 1984; Wilensky, 1984; Charniak et al, 1980). The article by Fikes and Kehler is a good introduction to KEE, however, for more detailed information, the reader is referred to IntelliCorp, of Mountain View, CA. (Fikes and Kehler, 1985).

The Fighter Pilot / System Interface

This section will describe the system's human-interface features. The primary user interface is the two monitor screens. They display appropriate information in various forms; for example, mission parameter panels, terrain profile views, and warning panels. Absolute minimum user keyboard interaction is required. It is imperative to exploit the mouse and menu capabilities, which, significantly, simplifies system use.

Mouse and Menus. Keeping the system simple to use is an extremely important consideration. Menus guide the user through the system hierarchical environment. Menus, typically require a single character for function selection. Menus relieve the user's burden of remembering exact command syntax and their associated order. This decreases the

dependence on the keyboard. The mouse, combined with menus, dramatically reduces the dependence on the keyboard. The program, being menu driven, does not require the pilot to have knowledge about the computer's operating system(s).

The Symbolics mouse has three selection buttons that can be programmed to send six different codes to the mouse processor, which is waiting to decode the input. The six different codes are single selection of a button L, M, R, and double selection of the button 2L, 2M, 2R. The mouse has an associated process that keeps track of its screen position. At the bottom of the monitor is a long darkened line with text displayed on it; this is the mouse documentation line. The documentation line is used to guide the user, by displaying system usage information. It also describes the functions available for each depressed mouse button. Each window can define its own menus and functions, which are mouse selectable. As the mouse travels over a window, which has defined mouse functions, the available options and instructions are displayed on the mouse documentation line. The user reads the instructions and depresses the appropriate button, displaying the desired menu. The user moves the mouse, pointing to the desired menu item, reads the documentation line, and makes his selection by pressing the correct button. This prototype relies on the mouse and window menus to simplify system operation. The user simply points the mouse cursor to a window, or at a

menu item, and 'shoots'; presses the appropriate button. There will be numerous examples, including pictures, detailing the use of the mouse and menus throughout the remainder of this chapter.

<u>Mission Parameter Panels</u>. This prototype currently uses two main panels located on the color monitor (see figure 4).

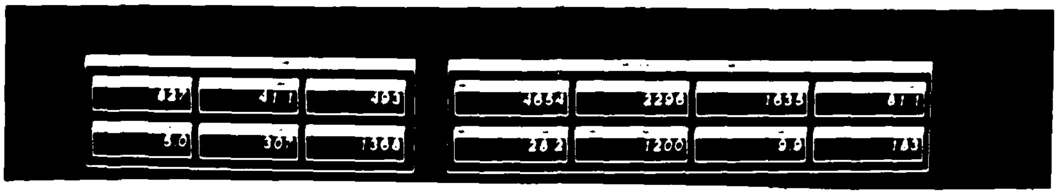


Figure 4. Mission Planning Parameters' Panel

The 'Low Level Leg Parameters' and 'Mission Critical Parameters,' panels display relevant planning information to the pilot. These KEE panel images contain KEE ActiveImages, which display mission values. As the pilot selects his turnpoints, or moves existing legs, critical mission parameters, such as, leg time, fuel used, leg distance and heading, point altitude, and total fuel, time, and distance remaining, are automatically updated. Appropriate conventional formulas and algorithms provide this numeric computation. This designed feature relieves the pilot the drudgery of explicitly computing them, allowing him to continue concentrating on mission level planning tasks. The pilot does not have to consciously track all the values, since they are continuously updated and displayed.

A strong 'quick prototyping' KEE feature is the ability to rapidly delete and create ActiveImages, including attaching the image to a particular object slot. Thus, if some pilots do not use a specific displayed value and would like to have other values displayed, the unused image can be deleted and replaced with a more appropriate customized one. This can be accomplished within twenty minutes, assuming the desired value already exists as an object slot. If the desired value does not exist as a slot, that is, an existing property of a mission object, the slot can be quickly created and added to the knowledge base within minutes. However, the availability and complexity of the function, which calculates the desired value, will determine the time required to complete this task. The topic of mission objects, their slots, and slot properties, will be described in detail in the 'Knowledge Database' section.

<u>High Level Cost Function Concept</u>. The 'Mission Critical Parameters' panel displays four important values: the distance, time, fuel used, and heading from the currently proposed location indicated by the mouse arrow, direct (straight line) to the IP. These computations use the fighter speed displayed in the 'Leg Speed' image window, located in the leg parameters' panel. These values represent the minimum cost of further mission planning. If the cost of flying straight to the IP exceeds the resources available, further detailed planning is not necessary until

these high level constraints have been satisfied by modifying the earlier portion of the plan. This point will be demonstrated in our example. This cost function concept is analogous to those associated with search algorithms designed to prune the search space; such as, the A* algorithm (Nilsson, 1980:76-79).

The Knowledge Database

The prototype knowledge base comprises numerous objects, or frames, referred to as KEE units (see figure 5).

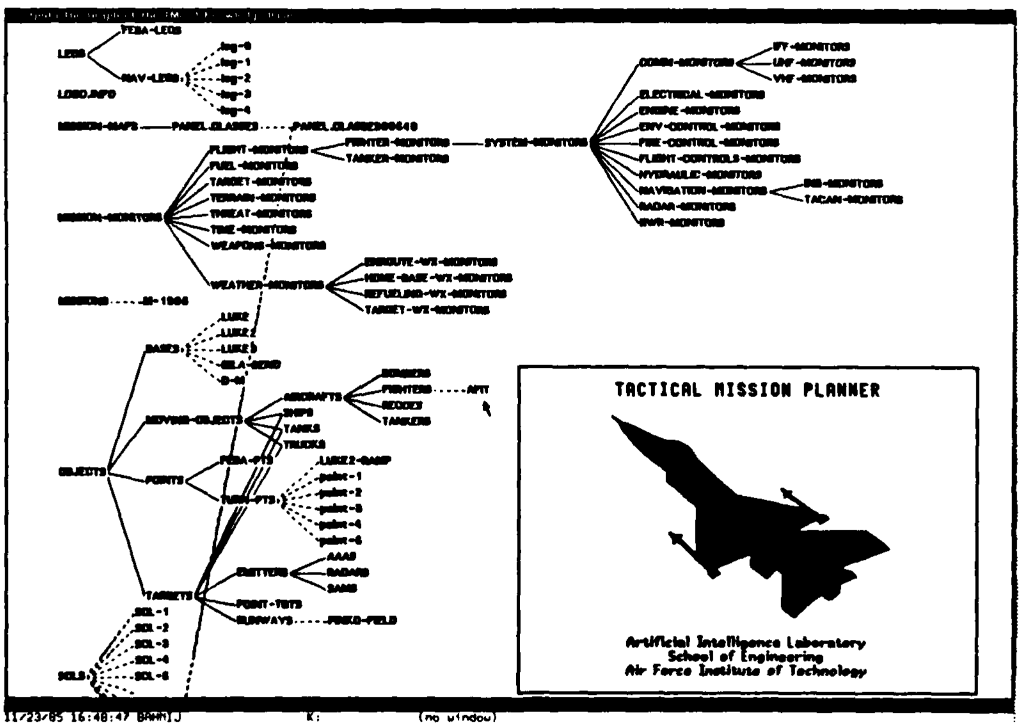


Figure 5. Tactical Mission Planning Knowledge Base

Each object has a set of slots, which contain various types of information, such as specific values or procedures, that

define the object's behavior. The information contained in these slots, along with their interaction, represents the domain knowledge. There are two types of slots; member slots, which can be passed down to subclasses, and own slots, which remain with its particular class unit or template. Passing slots down the hierarchical chain is called inheritance of properties from parent slots. Figure 5 displays the Tactical Mission Planning (TMP) object taxonomy, representing the domain knowledge database.

An object is the description of individual items or class of individual items, which is represented as a frame, called a KEE unit. These units are only templates of domain objects, when a specific object is created, that is, instantiated from a class, a mission element is activated. The ATO creates the fighter, AFIT, from class 'FIGHTERS' and assigns it a specific mission. AFIT acquires, inherits, all the FIGHTERS' member slots, which are now filled with the appropriate specific values.

Although KEE units have powerful facilities for describing frame attributes and behaviors, only a few basic features will be discussed. A slot can hold several types of information, a specific value, such as a pointer to another unit or data, or a method. These portray declarative and procedural knowledge representation schema, respectively. Methods are lisp coded procedures used to determine dynamic domain values. The concept of embedding procedures

into slot values is typically referred to as procedural attachment. When a function needs AFIT's speed for input, it sends a message to AFIT's slot, named 'SPEED.' Since aircraft speed is dependent on the its location and phase of flight, this message triggers the procedure, attached to the speed slot, to compute and return the value for current speed. A second form of procedural attachment is by use of KEE active values. Active values are a set of production rules, a procedure, that is activated, or triggered, anytime the slot value is accessed or stored. Active values represent a vehicle for updating the values displayed in the mission panel windows, described previously. Thus, active values behave as demons, supporting information hiding.

AFIT is a specific instance of the KEE class unit FIGHTERS. The slots chosen to describe AFIT are just one possible set of object attributes, which allow the designer to model system dynamics. These slots were created incrementally as the need for certain information, to describe object behavior, arose. To familiarize the reader with TMP KEE objects, the 'AFIT' unit will be examined in detail.

AFIT represents a specific F-16 fighter aircraft, assigned, in the ATO, to destroy the enemy airfield, 'PINKO-FIELD.' Figures 6 through 9 present the frame-based representation of the AFIT unit. The top window displays a portion of the hierarchical knowledge database. The bottom two windows display the slots that comprise the AFIT unit.

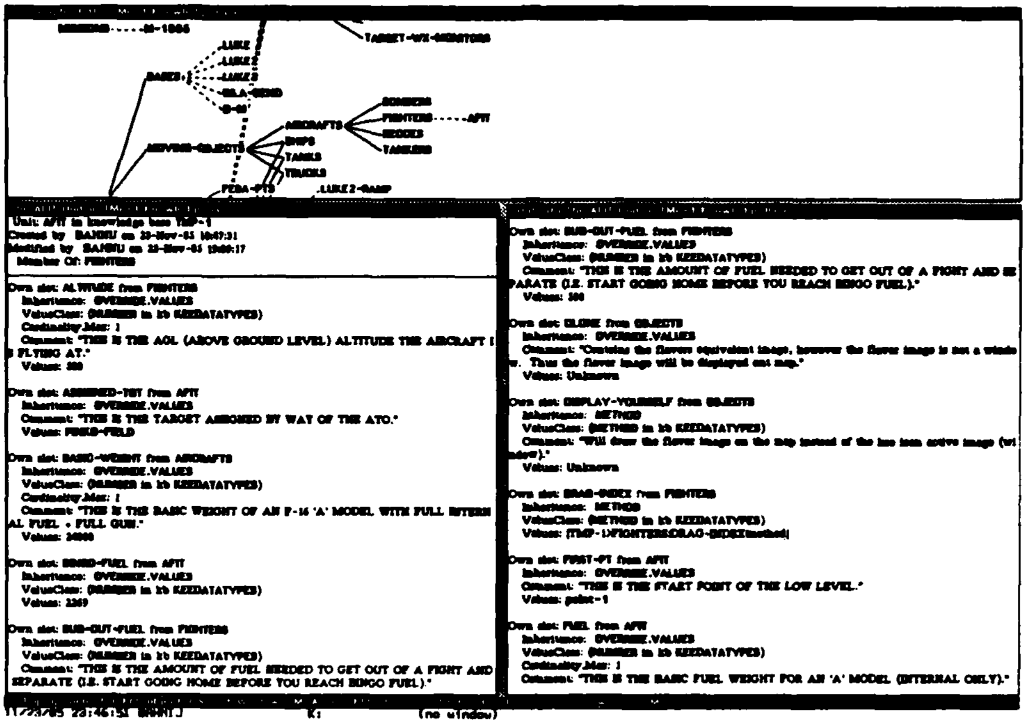


Figure 6. AFIT Frame Taxonomy

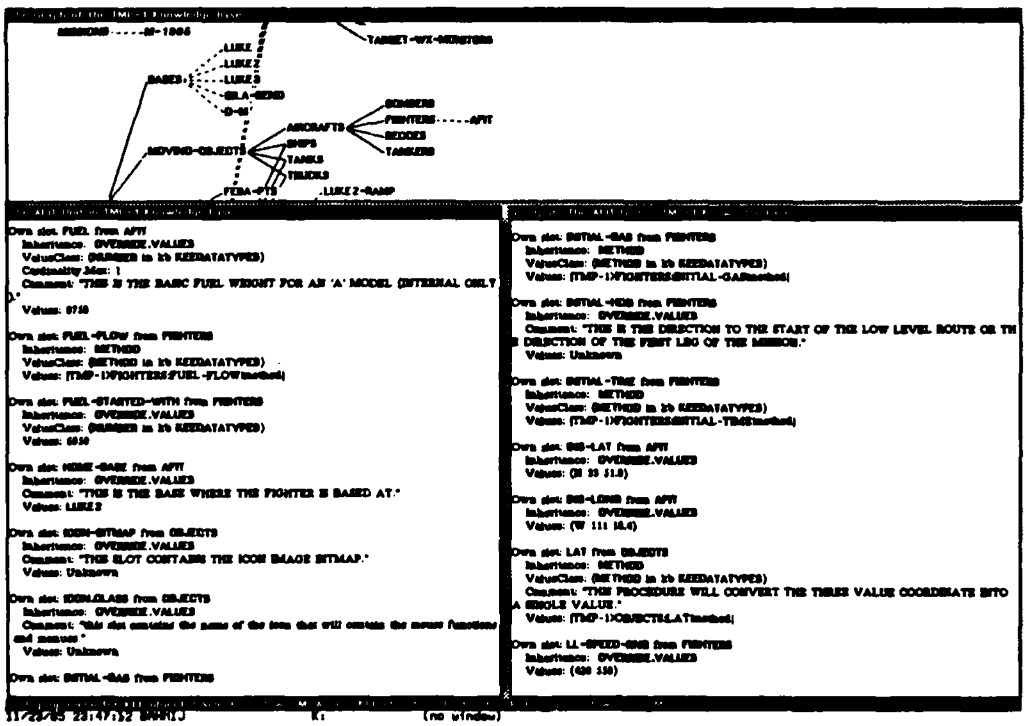


Figure 7. AFIT Frame Taxonomy (continued)

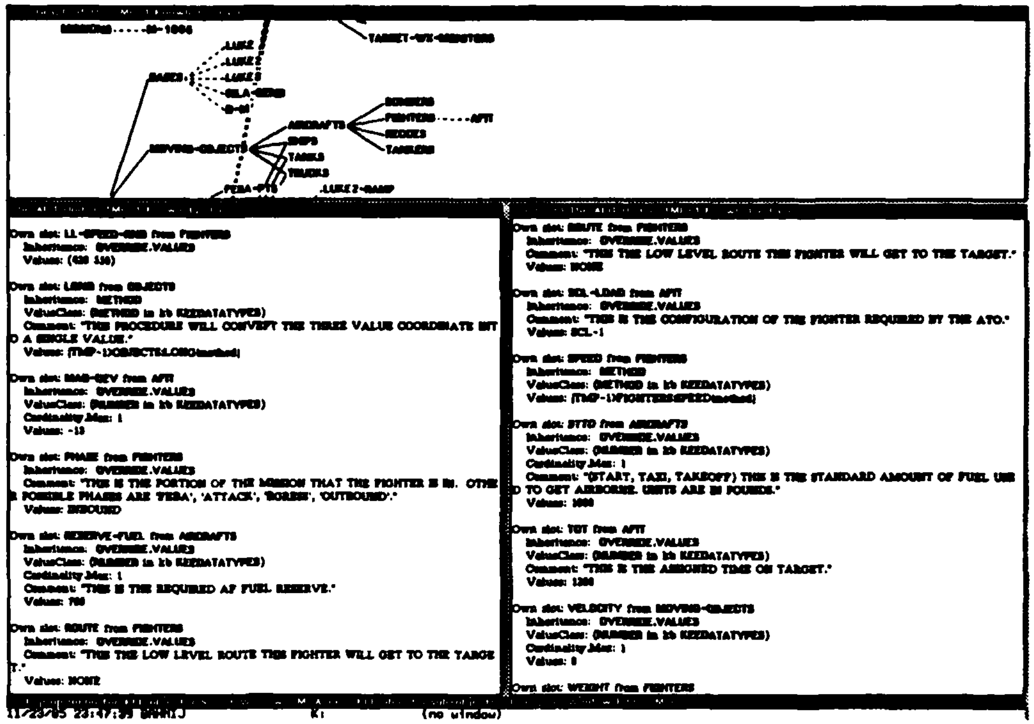


Figure 8. AFIT Frame Taxonomy (continued)

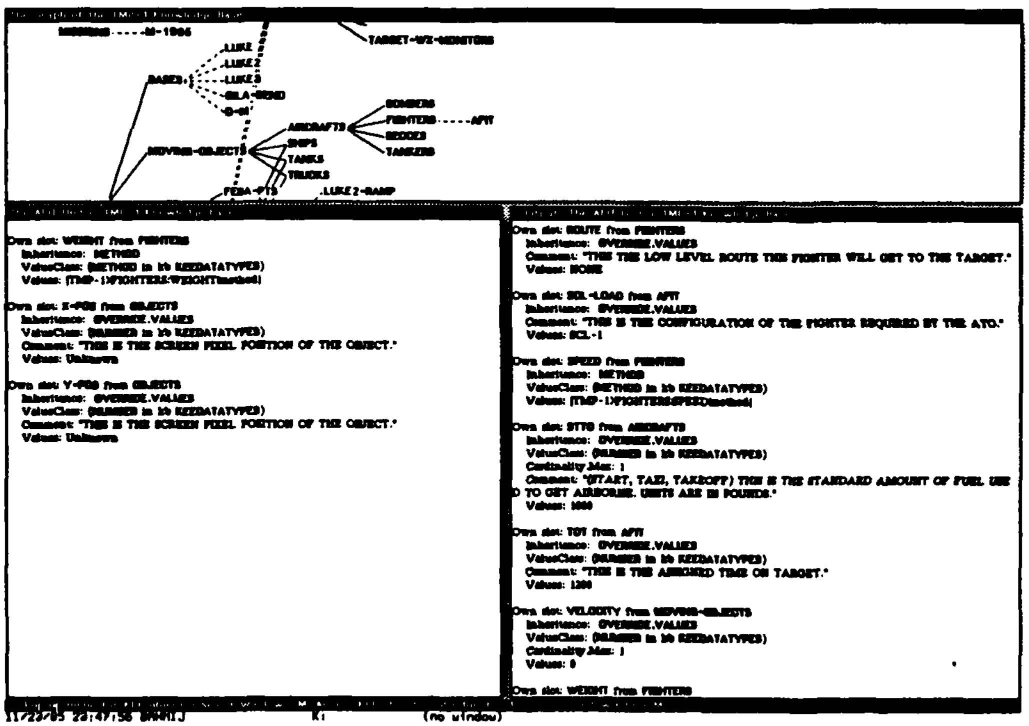


Figure 9. AFIT Frame Taxonomy (continued)

The **ALTITUDE** slot contains the altitude, a single value, at which the fighter intends to fly, in order to avoid enemy detection. This altitude is measured in feet above the ground, not in feet above sea level or pressure altitude. The pilot can easily change this value by utilizing the mouse function on the 'Fighter's AGL (for view)' image window, contained in the ' Sam & Fighter (Scene & View) Parameters' panel located at the bottom left corner of the main screen. The Fighter's AGL image window is a KEE digiactuator ActiveImage; this means the value of the window can be changed by positioning the mouse arrow in the window; depressing and holding the left mouse bottom, while moving the resultant arrow up, to increase the value, or down, to decrease the value. This window is linked to the **ALTITUDE** slot of the AFIT unit. Changing the window value changes the appropriate slot value. This capability will be displayed later in the chapter.

The **ASSIGNED-TGT**, a static value, slot is input by the ATO. **BASIC-WEIGHT** is determined from the SCL value, which is input by the ATO. **BINGO-FUEL** is computed, once the mission environment is established, that is, the location of the target, home base field, and the FEBA. This function assumes the fighter flies at 540 knots, leaving the immediate target area, slows to 480 knots until reaching friendly territory; climbs to a more fuel efficient altitude (28,000') and further slows to a more fuel conserving speed

(400 knots). **BUG-OUT-FUEL** is a predetermined amount of fuel added to the bingo fuel to ensure the fighter will depart the target area in time. The sum of bingo fuel and bug-out-fuel is called the joker fuel.

CLONE is a pointer to the Flavor's icon image, which is used to graphically display the unit image on the map. The link between KEE objects and Symbolics flavor objects was required because of the inappropriate KEE window icon image implementation. The KEE unit object contains the domain knowledge and the Symbolics Flavors clone object is used as the graphic image. This point will be clarified in the mission planning functions section, later in this chapter. The **DISPLAY-YOURSELF** slot contains a lisp function (a KEE method) which bitblt's its appropriate icon bitmap onto the desired map location. The location coordinates are contained in the **INS-LAT** and **INS-LONG**, as well as **X-POS** and **Y-POS** slots. **DRAG-INDEX** contains a lisp function which computes the aircraft drag index, a value used in computing current fuel flow. This function determines the fighter's current configuration, that is, the type and number of external stores the aircraft is presently carrying. The remaining slots comprising the AFIT unit are displayed in figures 7 through 9. Slots with 'Inheritance' and 'Value-Class' facets of METHOD contain a lisp function, otherwise they contain a specific value or values.

Mission Maps: The Planning Environment

Mission planning is accomplished by selecting functions and manipulating mission objects on a 'smart' contour map displayed on the main monitor screen (see figure 10).

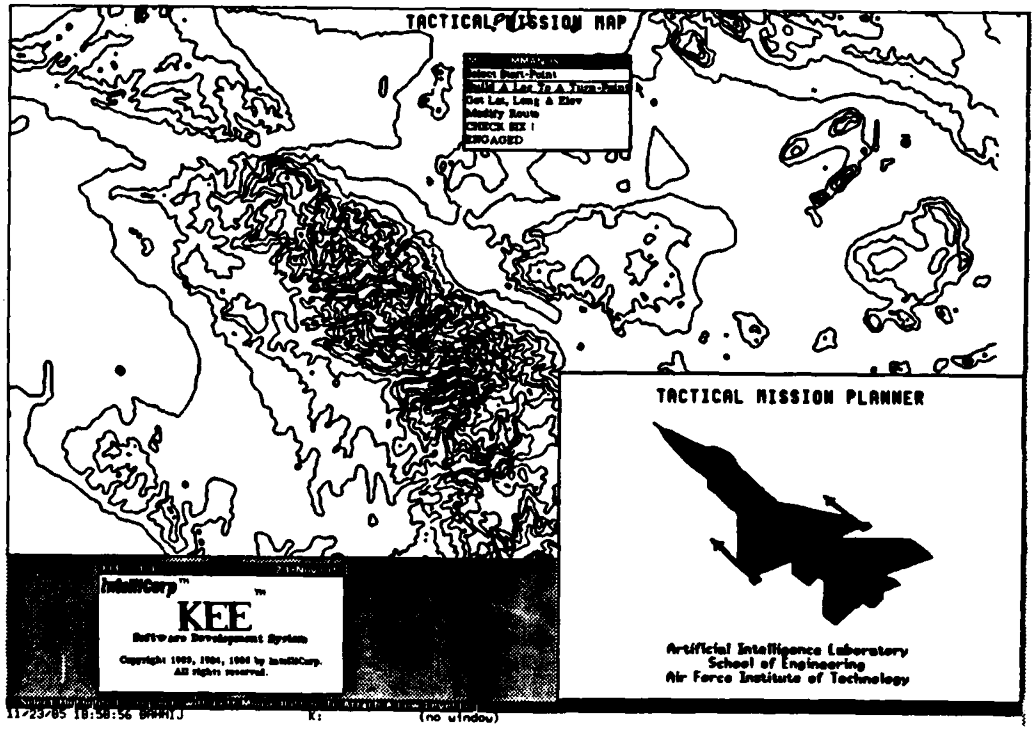


Figure 10. Mission Contour Map

The actual mission map is a KEE image panel which inherits properties from the generic KEE images unit and the domain customized PANEL.CLASSES unit. PANEL.CLASSES is a subunit of the MISSION-MAPS unit. The PANEL.CLASSES unit was required to build domain specific mouse selectable map menus and functions, topics to be covered in the next section. A pre-processed contour map containing the desired targets and home bases is selected, when the ATO is input, and bitblt'ed

onto the KEE panel, when the large scale mission map is mouse selected on the mission planner logo panel (see figure 11). This contour map works in concert with a sixteen bit array, containing DMA elevation data, and having the same dimensions as the displayed contour map.

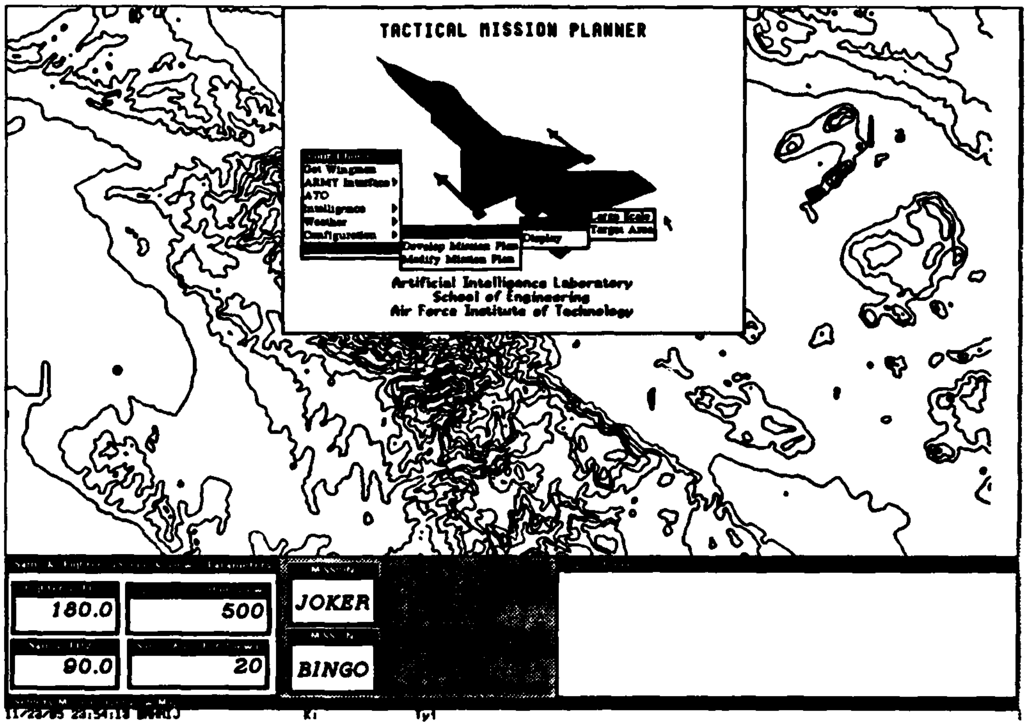


Figure 11. Mission Planner Logo Panel

DMA data is acquired from the Defense Mapping Agency, located in St. Louis. The elevation data, stored on VAX tapes as a 1201 by 1201 array, is organized in contiguous one degree squares (60 nautical square miles). An elevation reading is taken every three arc seconds storing the value, measured in meters. The data is first converted to feet, then the array is transferred to the Symbolics. This

elevation data array is used to pre-process a contour map, using a simple edge detecter algorithm to draw a contour line every 500 feet. The X and Y coordinates of a map pixel are used as indices into the sixteen bit elevation array, returning the point elevation. The Mission Planning Functions' section will present more information on the selectable functions associated with the mission map.

Constraints

The ATO inputs the major high level factors that constrain the mission planning process. This is accomplished by mouse selecting 'ATO' on the logo panel, reference figure 11. The TOT, target location and SCL establish the time, distance, and fuel available values bounding the environment. Intelligence and weather factors, such as, threat locations and bad weather areas, indicate geographic areas the pilot should avoid flying, if possible. These factors are also input via the logo menu items of 'Intelligence' and 'Weather.' Keeping track of these constraints, as well as their overlapping interactions, is a complex and difficult task. Norman addresses this particular topic, in detail, and presents an algorithm for examining the overlapping complexities, from a distributed processing perspective (Norman, 1985). The chore of constantly monitoring these critical factors can be off-loaded to the computer. The pilot will now be advised when any of these factors crosses a predefined threshold. This 'management by exception'

approach mission planning allows the pilot to successfully function in the time limited, complex tactical mission planning environment.

<u>Constraint Monitoring</u>. The 'Mission Critical Parameters' panel, discussed earlier, represents a typical set of planning factors. The ActiveValue KEE feature is used as a demon embedded in a particular slot, performing a procedure warning the pilot when the slot value constraint is violated. Any slot in the knowledge base can have an ActiveValue, demon, attached to it. Every time the slot is accessed, changing the value or just consulting it, the ActiveValue procedure is triggered. The system designer can define a procedure or set of procedures, which the demon will call to either warn the user of a conflict and/or propose solutions to resolve the conflict.

<u>Constraint Violations</u>. When constraints are violated the pilot is warned of the mission planning conflicts. The pilot can immediately modify the plan, bringing the relevant parameters within acceptable range. The pilot also has the option of relaxing the specific constraint, an example previously discussed. Constraint violations can be flagged in different ways. The 'JOKER' and 'BINGO' fuel alarm panels, shown in figure 11, are one method. When the value of the mission fuel remaining slot, belonging to the AFIT unit, drops below the joker level, a message is sent to the joker alarm, causing the alarm image to flash. When the fuel

value falls below bingo the joker alarm is reset and the BINGO alarm is triggered. Displaying messages on the screen is another form of alerting the pilot of planning conflicts. The demon can also send an appropriate message to a conflict monitor and/or resolver knowledge source (module or KEE unit). This unit contains rules and procedures (domain heuristics) used to resolve the conflict. These proposed solutions are presented to the pilot for consideration and evaluation. The following example will clarify this point. The normal active runway used for takeoff at Luke AFB, AZ, is runway 35; 35 meaning, takeoff heading, 350, North. If the start point of the low level navigation phase is located to the south of Luke, taking off heading South, using runway 17, would save two to three minutes and approximately four to six hundred pounds of fuel. The surface winds at the airfield have to permit opposite direction takeoffs. A tailwind component of less than ten knots is the typical heuristic allowing this option. Another form of conflict identification and resolution will be described later in the chapter.

Mission Planning Functions

Once the appropriate mission map was selected by mousing the TMP logo window and selecting the 'Large Scale' menu item, described earlier and shown in figure 11, the planning process can begin. The TMP logo window has to be moved, to make room for the mission map, soon to be displayed. Move

the mouse arrow onto the logo window and press the right button, displaying a KEE menu. Moving the mouse arrow to point at the 'Shrink' menu item and pressing the left button will accomplish this task. The logo window will change to a small aircraft icon image and move to the bottom of the screen next to the fuel alarm panels. This miniature icon window retains the original mission level menus and functions. To display the mission map, move the mouse arrow to the small aircraft, mouse left, move down the mission cascading menu and select 'Display.' The mission map will be displayed on the screen, including icon images representing the assigned fighter's home base, top right hand quadrant, and the target airfield, bottom left quadrant (see figure 12).

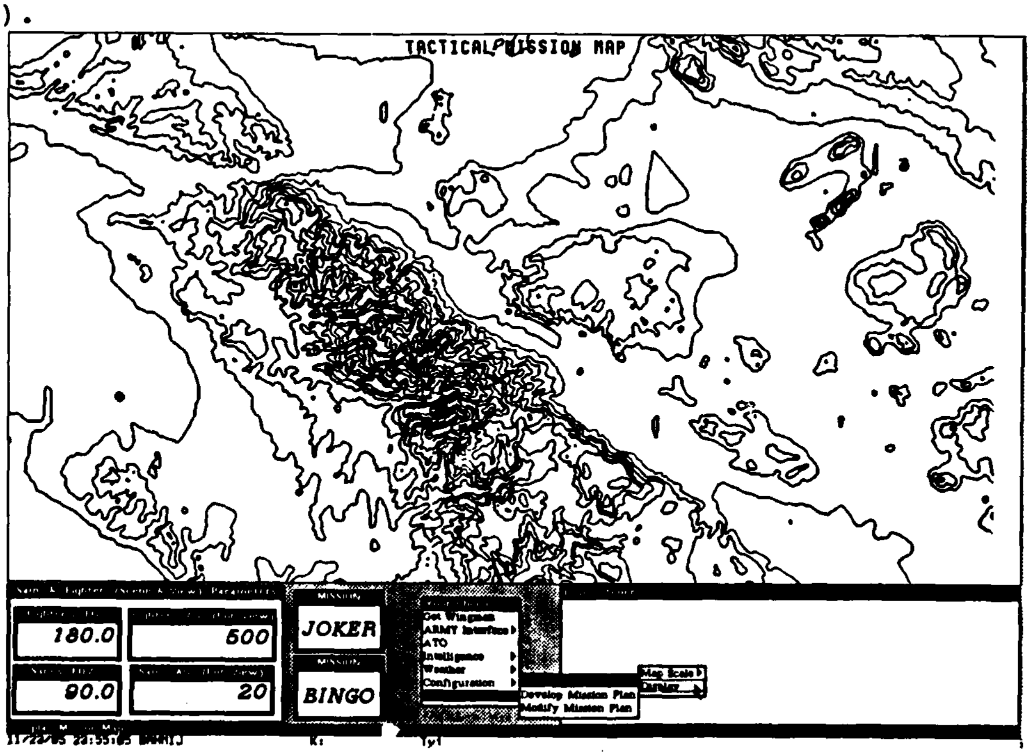


Figure 12. Mission Map

The entire map region is mouse sensitive. Currently, only the left and middle mouse buttons have been defined to access mission planning functions. The right button still presents the KEE right button menu. Selecting the start point for the low level navigation portion of the mission starts the planning process. Pressing the left mouse button, with mouse arrow anywhere on the map displays the 'Map Commands' menu. As you move the pointer to each menu item a brief function description and/or instructions are printed on the darkened mouse documentation line, located on the bottom of the main screen just above the date and time display. Figure 13 shows the result of selecting the 'Get Lat, Long, & Elev' item.

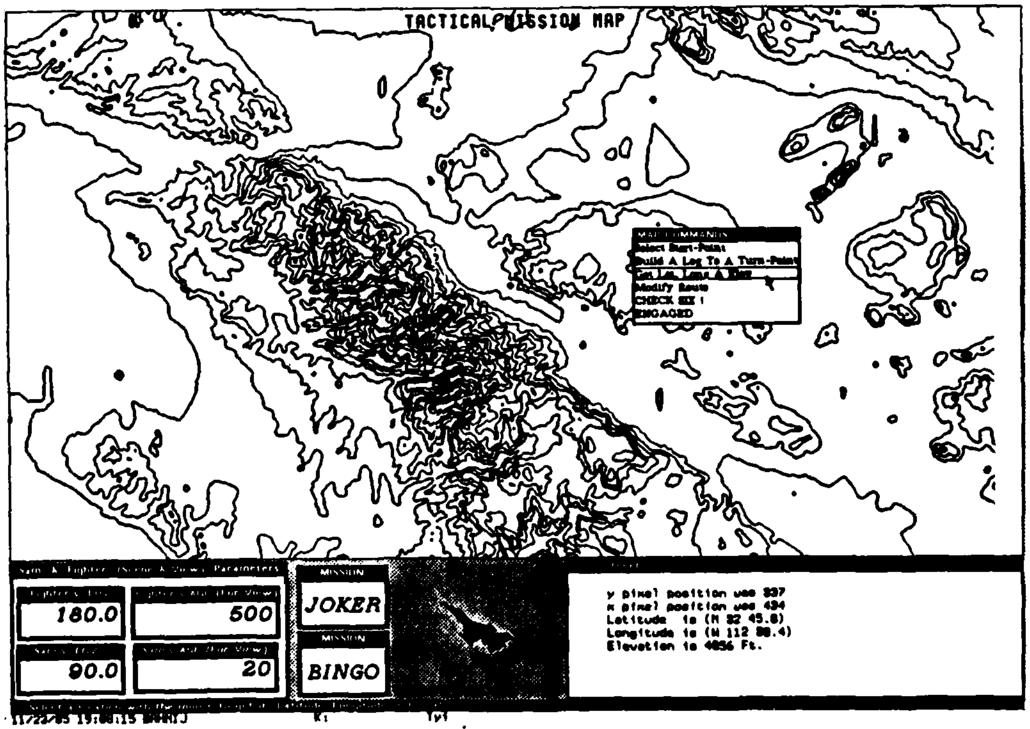


Figure 13. Latitude, Longitude, and Elevation Functions

The functions associated with computing the displayed information are used throughout the planning process. The latitude and longitude are displayed in F-16 usable formats.

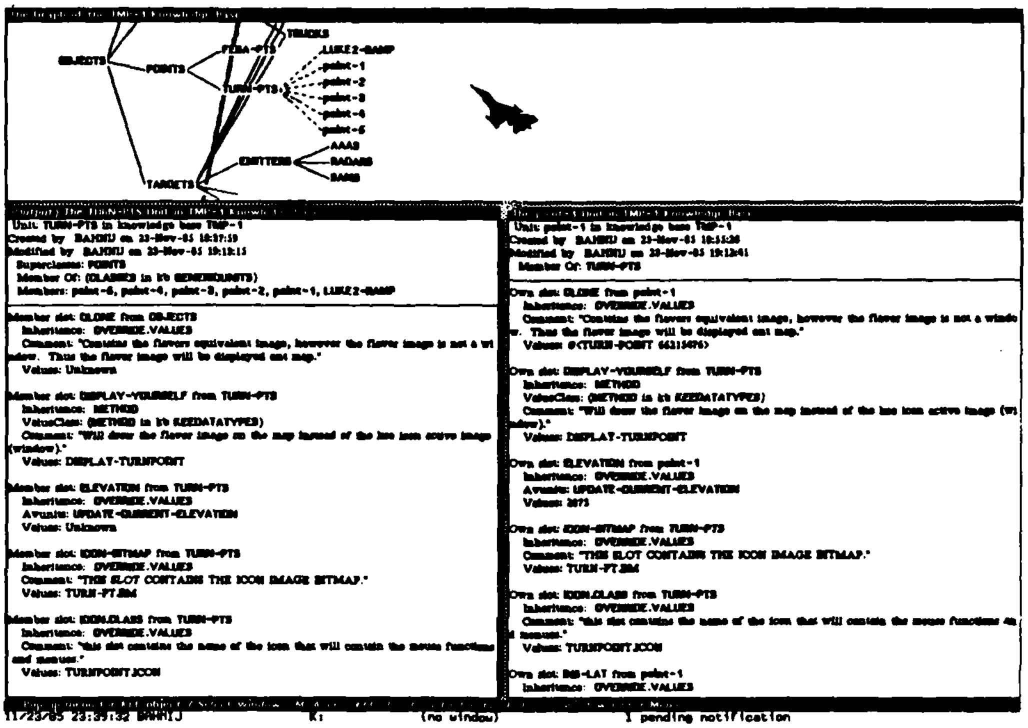


Figure 14. Point-1, An Instance of the Turn-Pts Unit

Selecting 'Select Start-Point' on the map commands menu instantiates a turn-pts, point-1, and a nav-leg, leg-0, unit (see figure 14 and 15). As the mouse pointer is moved on the map, looking for an appropriate location to start the low level, mission and leg parameter values are continuously updated and displayed on the color monitor screen. The initial value for mission fuel is 6950 pounds, the usable fuel for an F-16A configured with a centerline tank. The function for computing fuel used to the start point

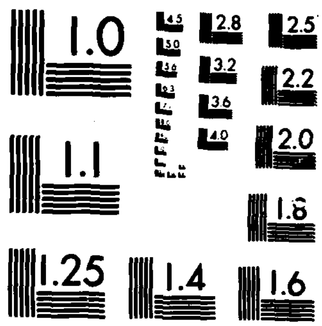
AD-A163 944 A FIGHTER PILOT'S INTELLIGENT AIDE FOR TACTICAL MISSION PLANNING(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING R B BAHNIJ DEC 85 2/2

UNCLASSIFIED AFIT/GCS/ENG/85D-1 F/G 6/4 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

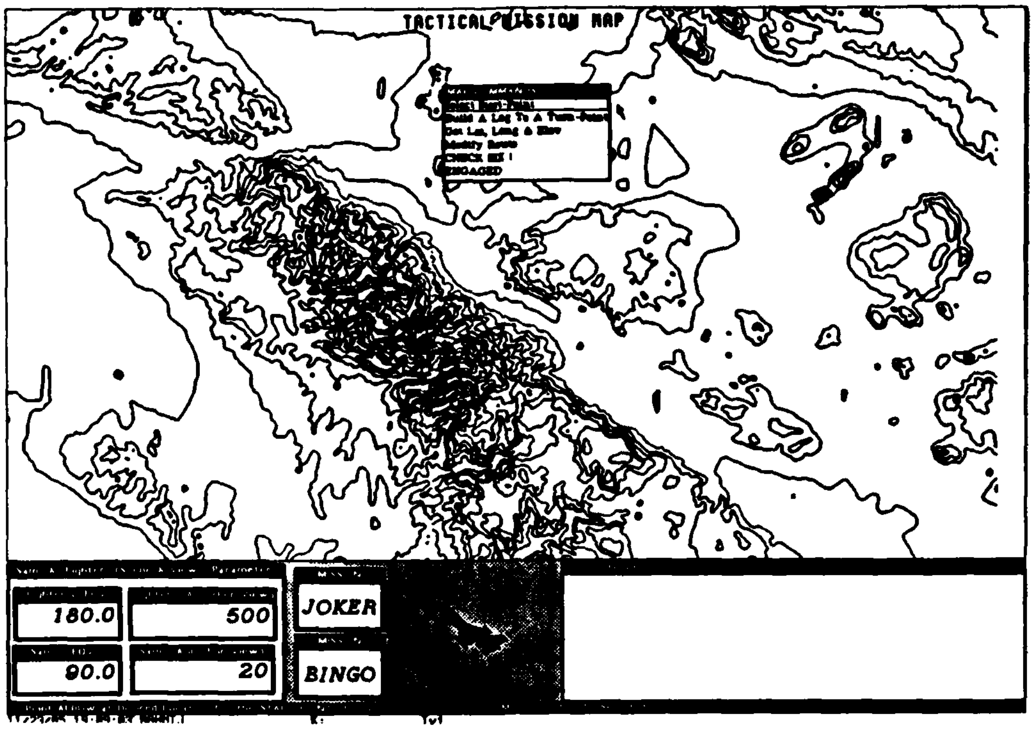


Figure 15. Select Start Point

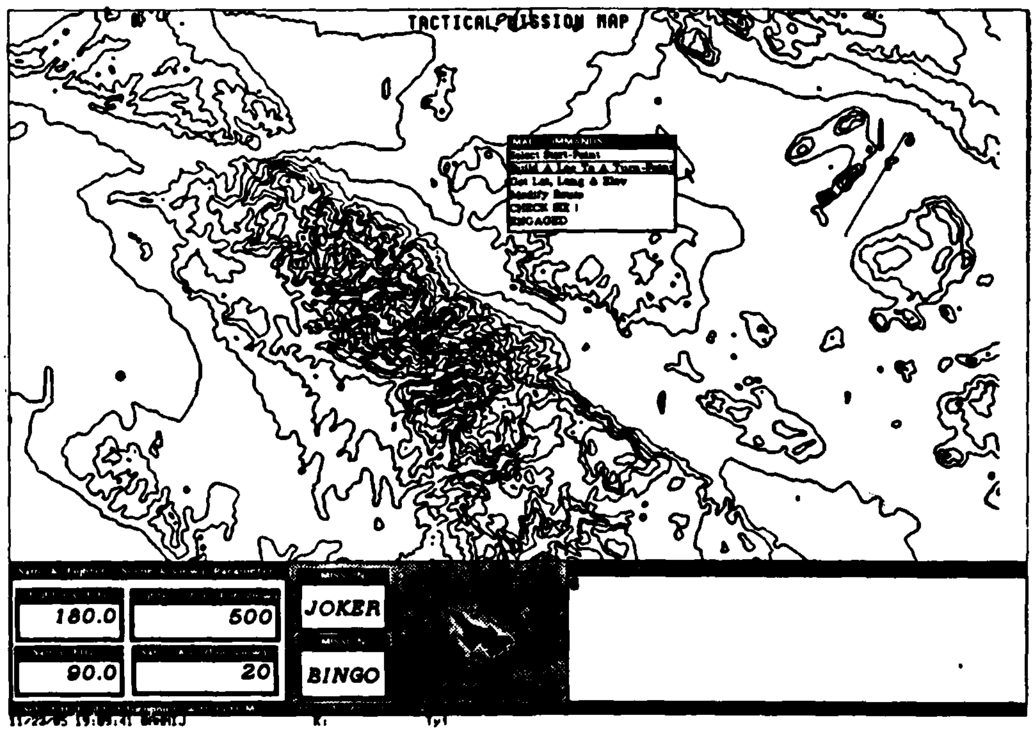


Figure 16. Build Low Level Navigation Legs

subtracts 1000 pounds of fuel, which accounts for 30 minutes of ground operations, an afterburner takeoff, and a 360 knot cruise to the start point. Once the start point is located, pressing the left mouse button selects that location and draws that flight segment (see figure 16).

The next step is to build low level legs, selecting 'Build A Leg To A Turn-Point' menu item, which changes the mouse pointer arrow from a slant to a vertical bold type. When this new arrow pointer is moved near the last turn-point, the turnpoint will highlight itself (see figure 17). Clicking left when a point is highlighted creates a nav-leg unit which is attached to that point (see figure 18). To abort after selecting to build a leg or modify the route, simply mouse left ensuring no objects are highlighted. Moving the new crosshair mouse cursor updates and displays the relevant parameter values. The fighter's speed value used in all appropriate calculations is displayed in the 'Leg Speed' image window on the low level leg parameters panel, located on the color monitor. This speed window is a KEE digiactuator panel, which means it can be modified in the same manner described for the 'Fighter's AGL' image panel. To move the mouse cursor from the main screen to the color screen, press the FUNCTION key followed by pressing the X key. FUNCTION X is a toggle moving the mouse cursor to and from the color screen. Once the mouse cursor is moved to the color screen, place it on the speed window, press and

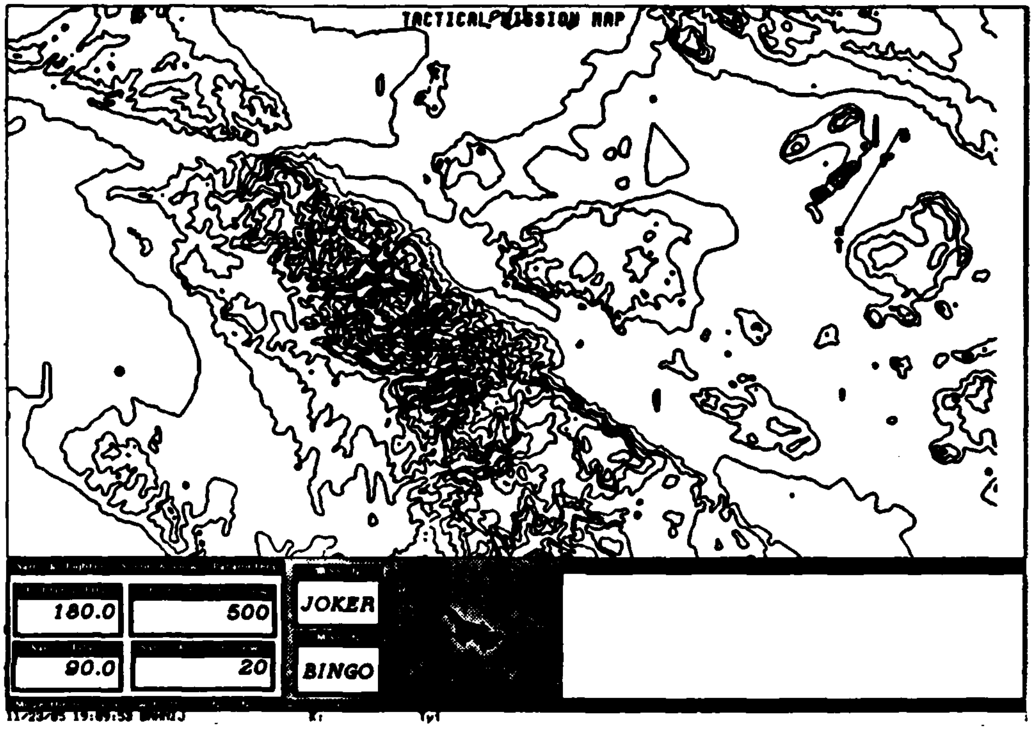


Figure 17. Highlighting the Turnpoint

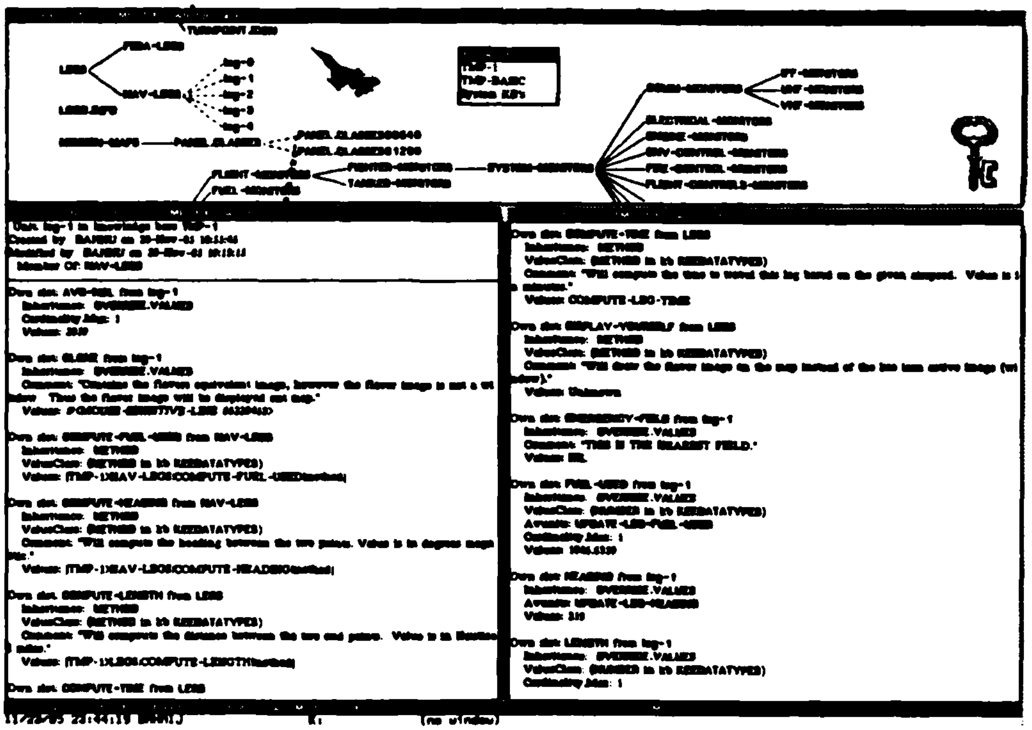


Figure 18. Leg-1, An Instance of the Nav-Leg Unit

hold the left mouse button, while moving the new arrow above or below the window. This will increase or decrease the window value.

If the fuel constraint is violated the JOKER or BINGO fuel alarm panels will flash, informing the pilot of the conflict (see figure 19 and 20). The pilot can modify the route to correct the fuel shortage problem, by selecting the 'Modify Route' menu item (see figure 21). Moving the vertical bold arrow near a turnpoint, highlights that point (see figure 22 and 23). Moving the mouse arrow near a leg will highlight that leg (see figure 24). Clicking left with an object highlighted selects it as the object to be modified. The remaining map functions are accessed through the middle mouse button menu, the 'Visual Display Commands.' The middle mouse functions perform two tasks. Displaying terrain profile views of proposed flight paths onto the color monitor allows the pilot to visualize topographic terrain features. Analyzing the impact newly discovered SAM sites have on the mission plan, is the second major feature. Prior to selecting either feature, always select 'Clear Display Area.'

Terrain Profile Views of Proposed Flight Paths

The selection sequence to display terrain profiles is 'Clear Display Area,' 'Place Fighter On Map,' 'Set Fighter's Look Direction' (see figure 25). The 'Zoom' function can be used after the profiles have been processed and displayed.

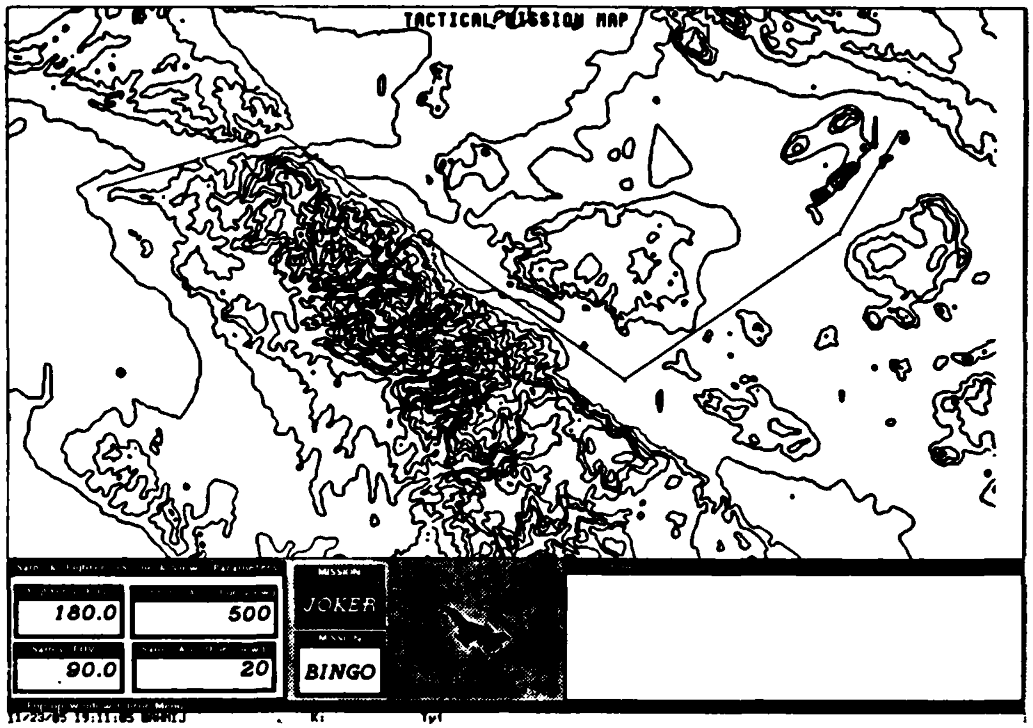


Figure 19. JOKER Fuel Warning

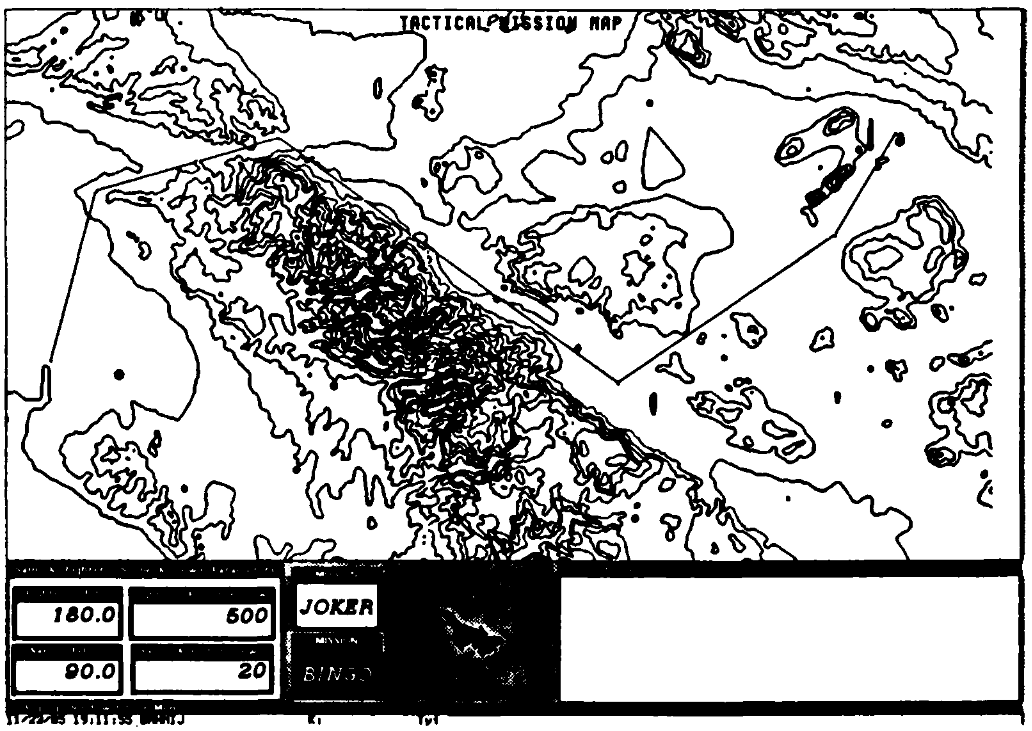


Figure 20. BINGO Fuel Warning

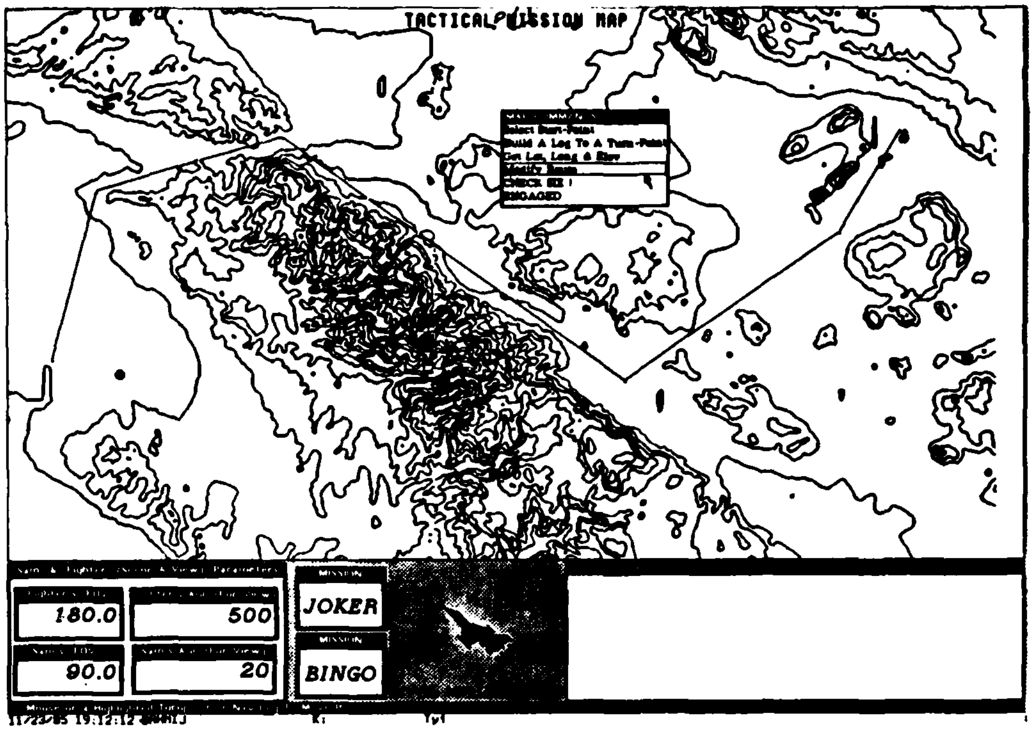


Figure 21. Modify Route

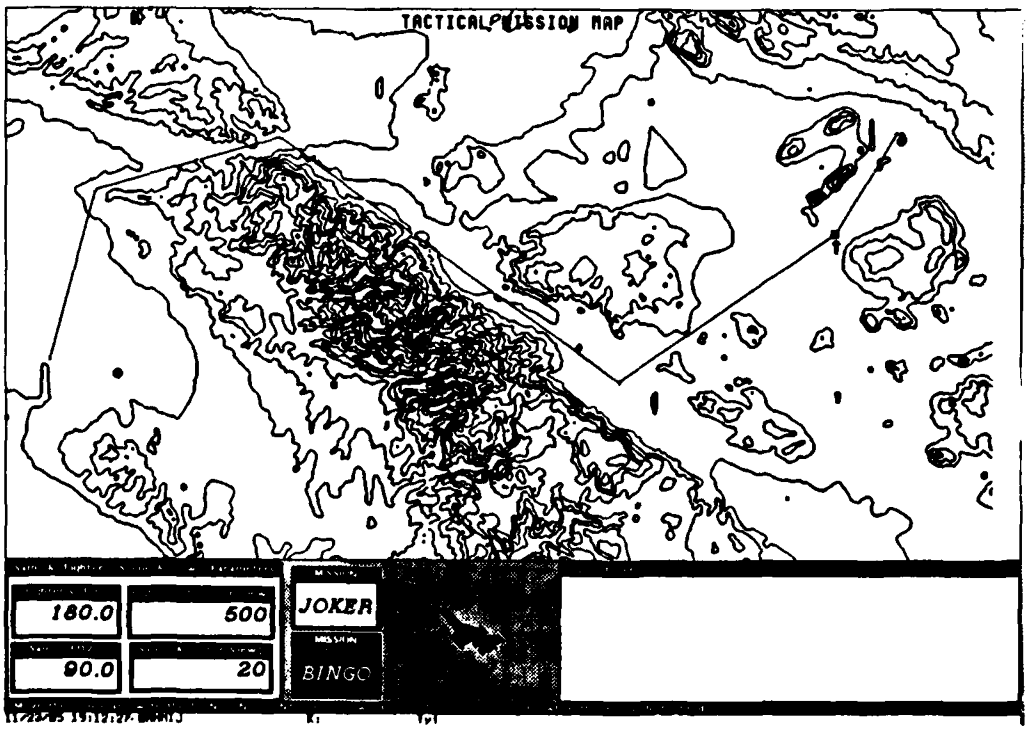


Figure 22. Select Point to Move

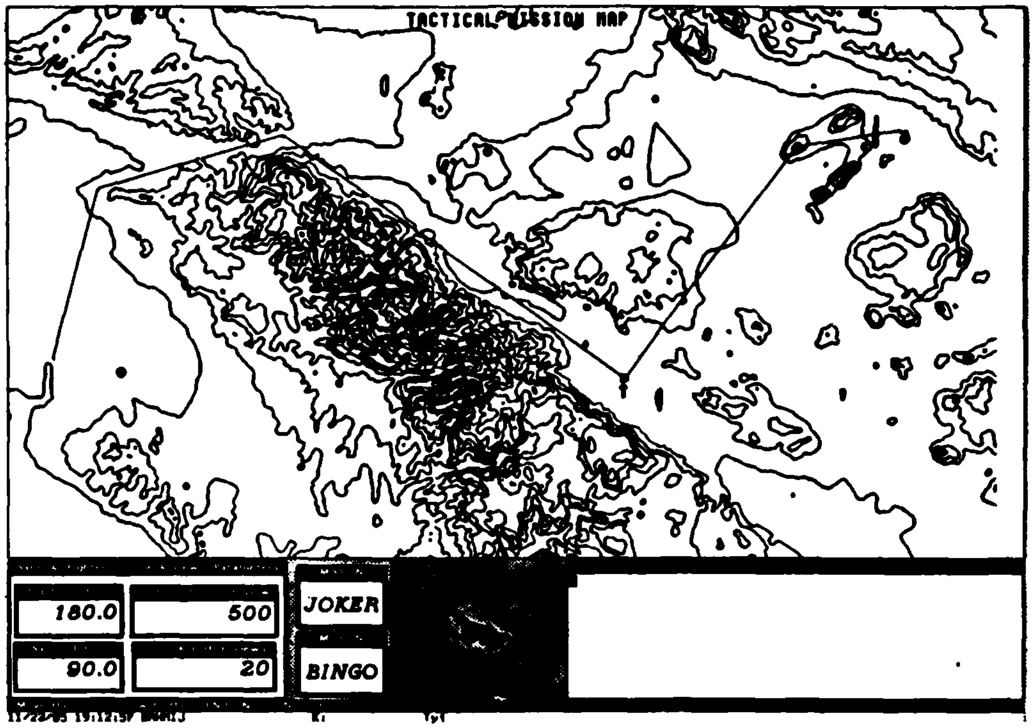


Figure 23. Select Another Point to Move

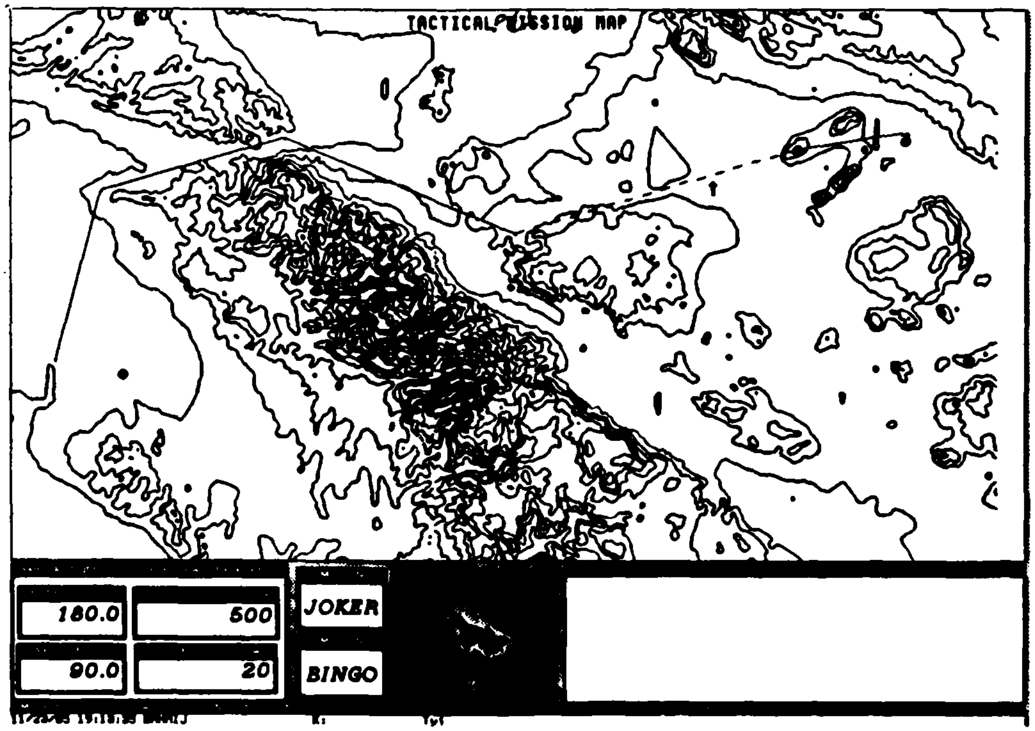


Figure 24. Select Navigation Leg to Modify

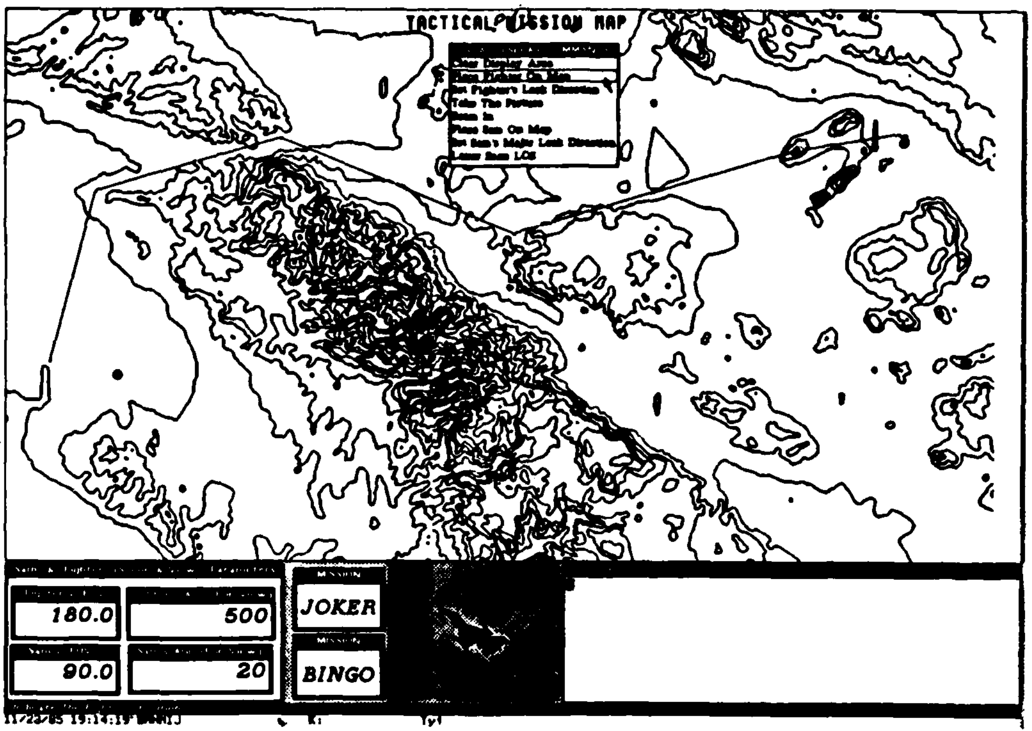


Figure 25. Terrain Profile View Menu

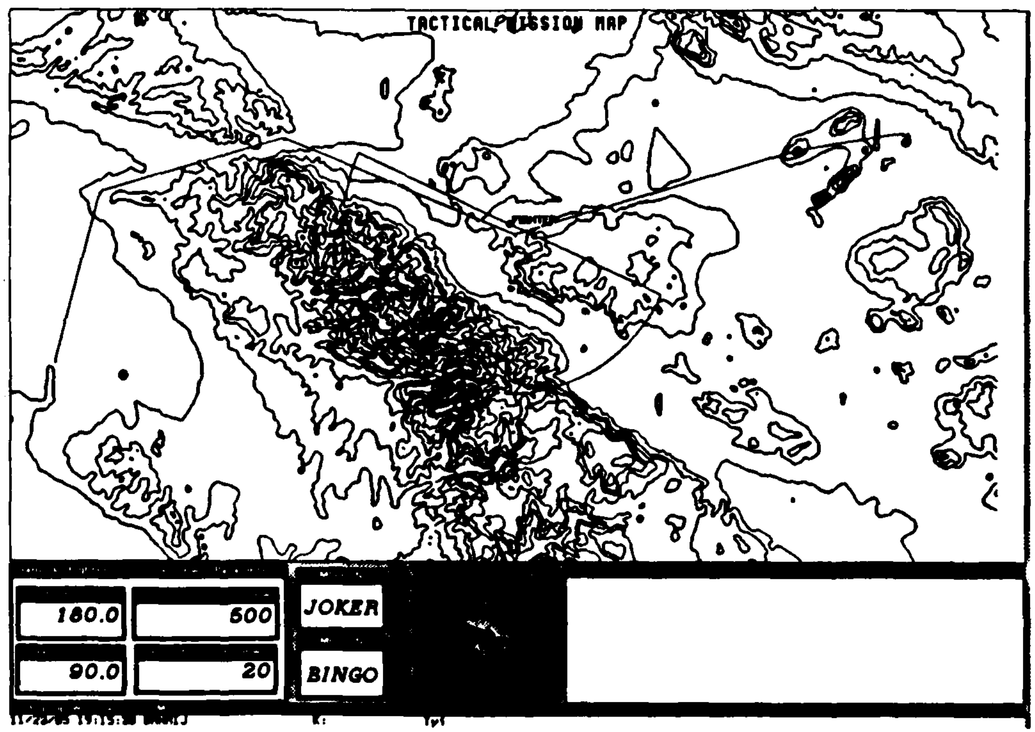


Figure 26. Fighter's Field of View

The pilot can select areas of the profile view to expand or zoom in on.

Selecting to place the fighter on the map, displays a crosshair cursor, used to place the fighter on the map. Mouse clicking left sets the fighter's position. The Fighter's field of view (FOV) can be set to any value from 0 to 360 degrees, by means of the 'Fighter's FOV' digiactuator. Before taking the picture the pilot selects and indicates the look direction (see figure 26). The center of the scan is the pilot selected fighter's geographic position. The vertical component used in the profile view calculations is input via the altitude displayed in the 'Fighter's AGL' digiactuator window, discussed previously. Figures 27 through 32 present the process, which is displayed on the color monitor. The pilot can elect to get a closer view of the terrain by selecting the zoom option. After zoom is selected, the pilot is querried for the zoom factor. A magnification of two or three is acceptable for current pixel resolution. Figures 33 through 35 depict the graphic sequence displayed by this process. The basic algorithm, developed by John Mitchiner and Laurie Phillips, at Sandia Labs, for autonomous land vehicle research, was modified by the author for the flight and surface to air threat domains (Mitchiner and Phillips, 1985). The basic approach used to display the profile views was applied to display and analyze threats.

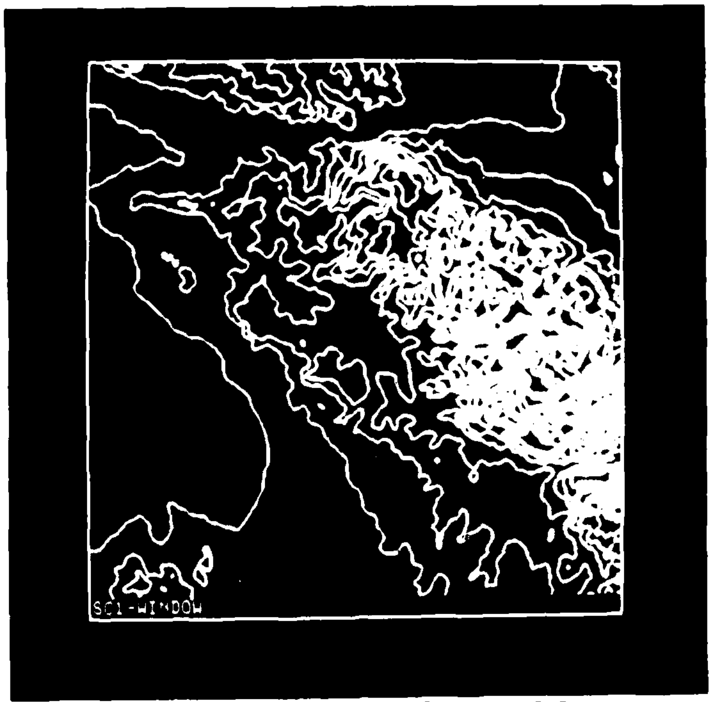


Figure 27. SC1-WINDOW

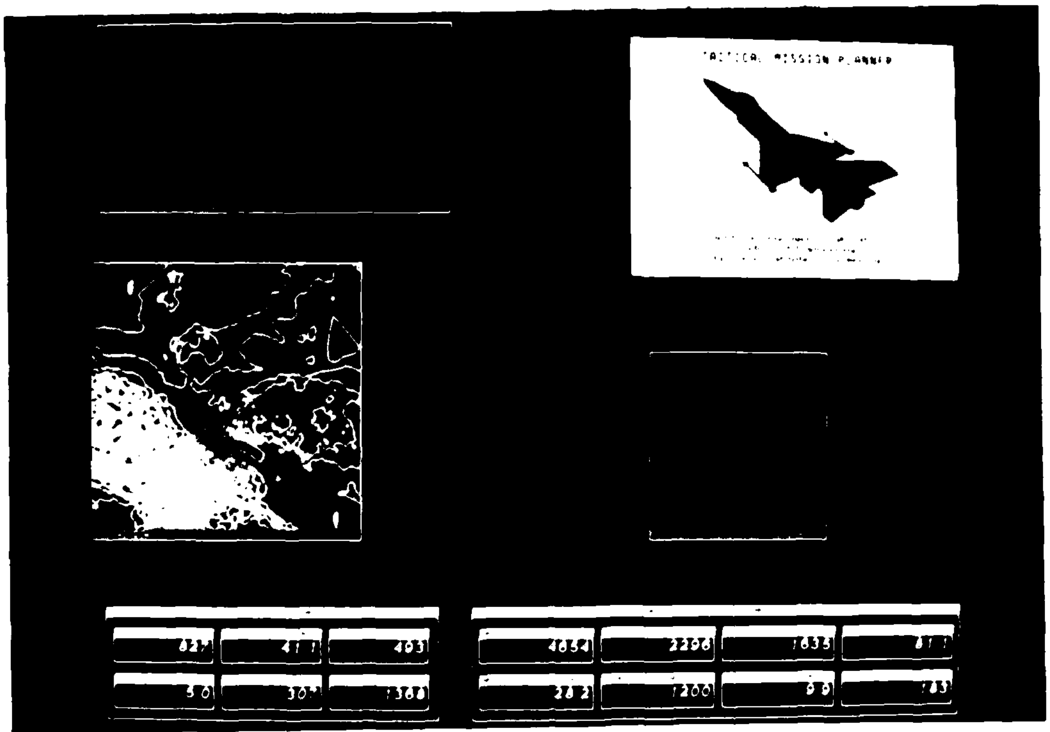


Figure 28. First 90 Degree Scan Complete

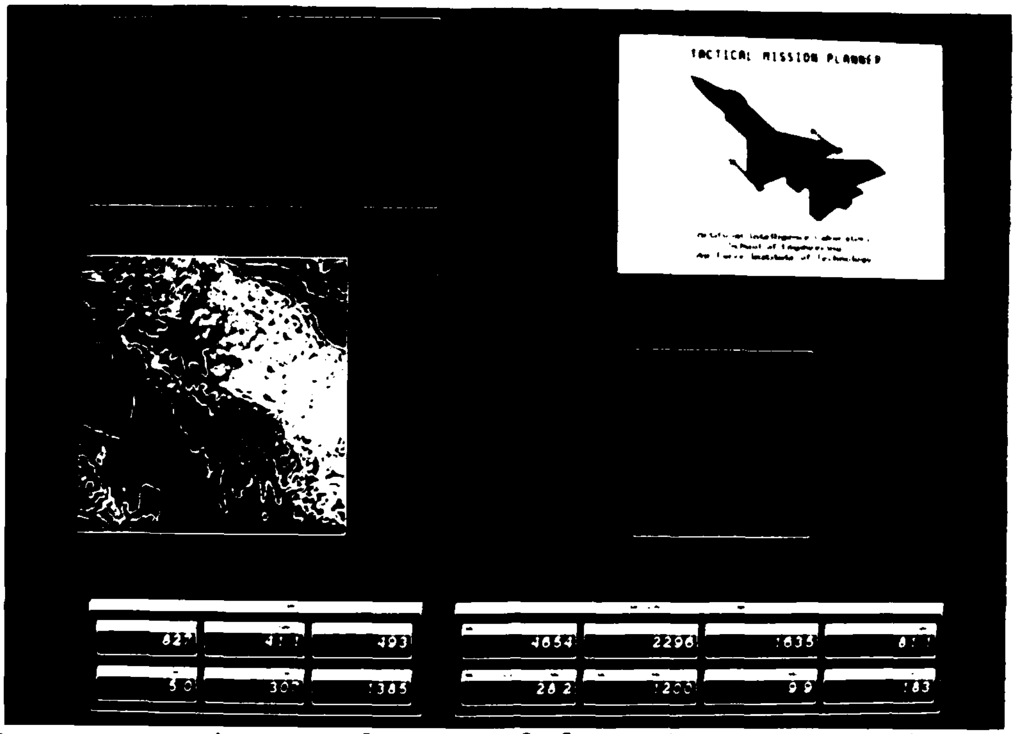


Figure 29. First Quadrant Scaled To FIGHTER-SCENE-1-WINDOW

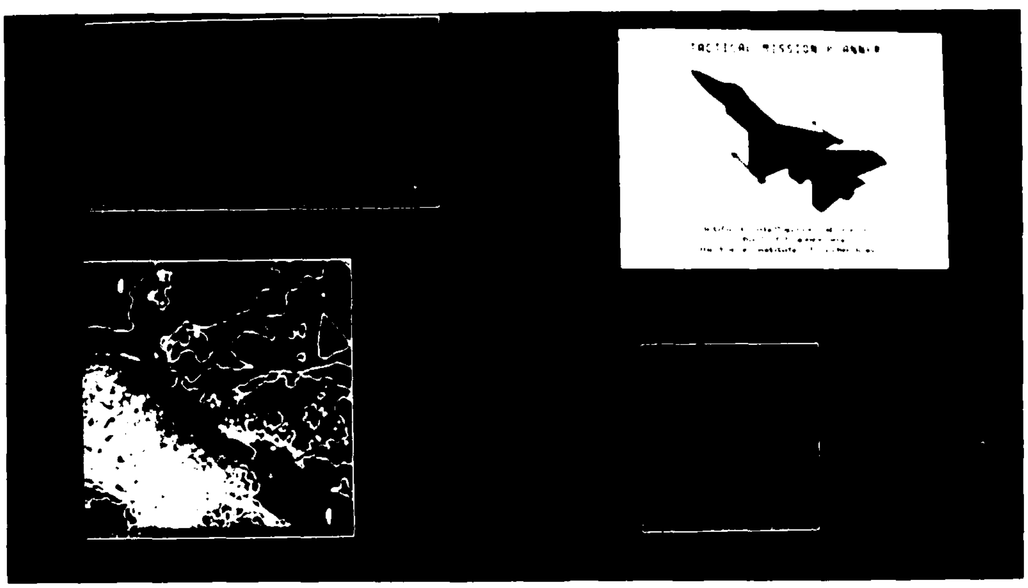


Figure 30. Second 90 Degree Scan Complete

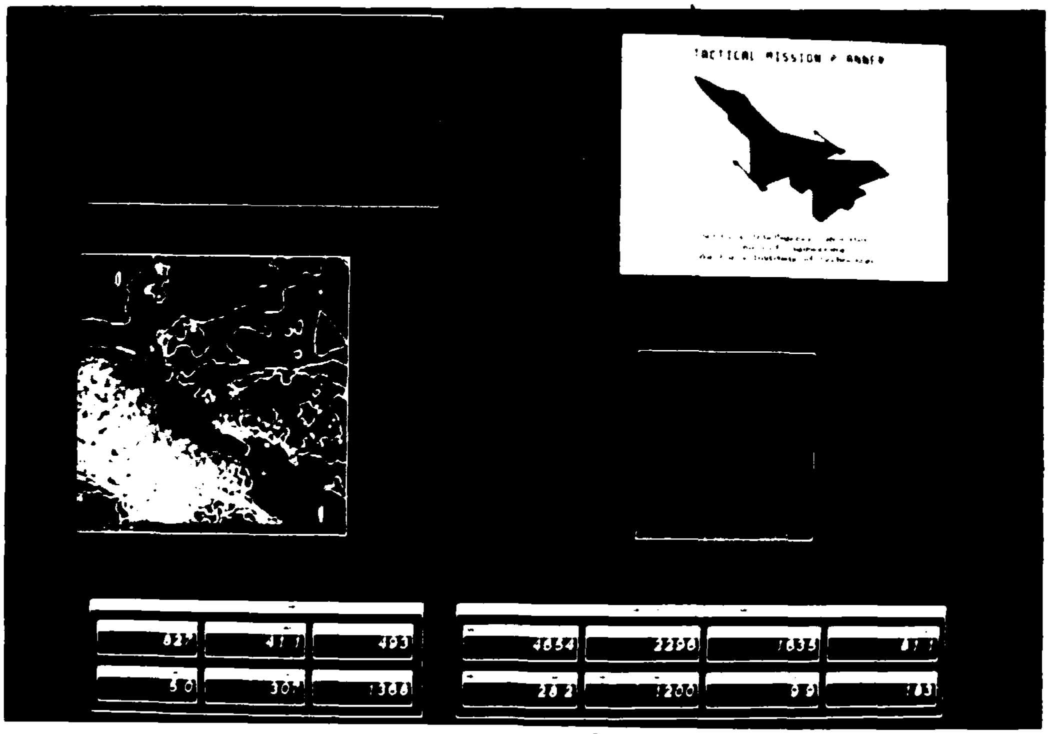


Figure 31. Second Quadrant Scaled To FIGHTER-SCENE-1-WINDOW

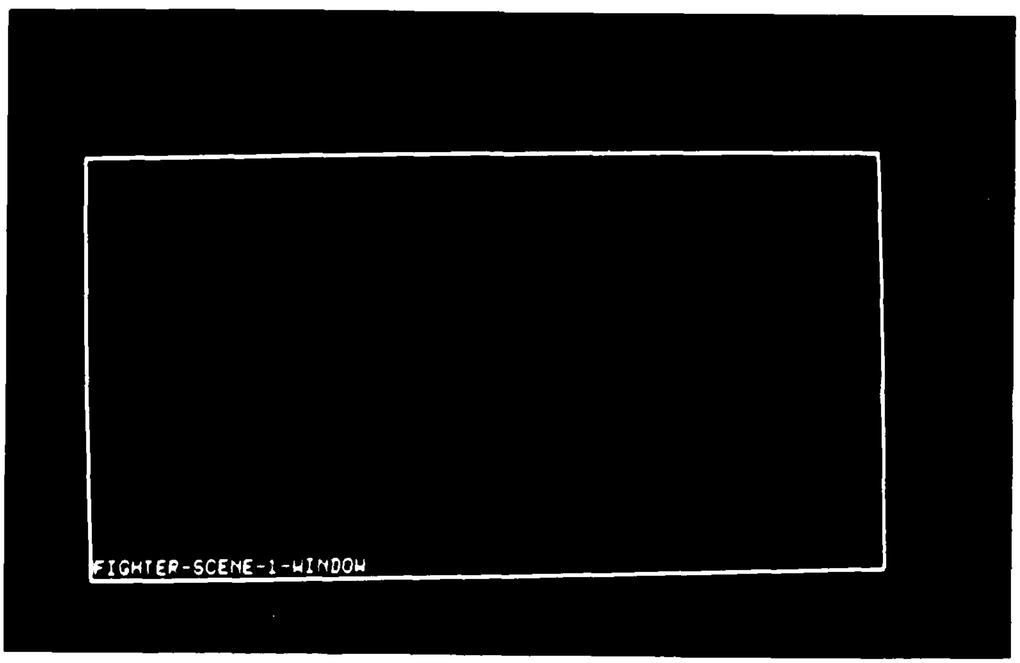


Figure 32. Terrain Profile View

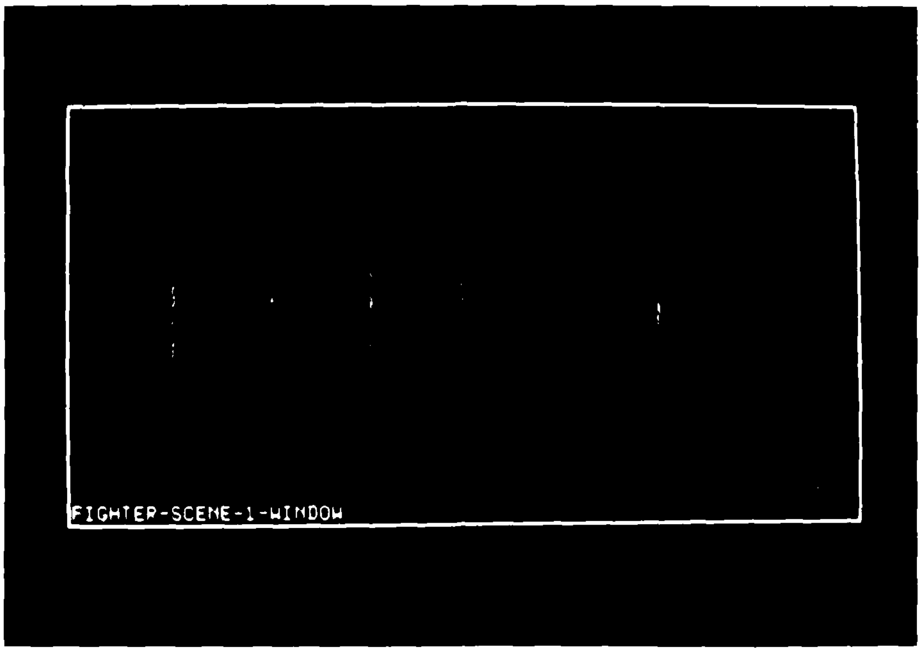


Figure 33. Zoom Points Selected

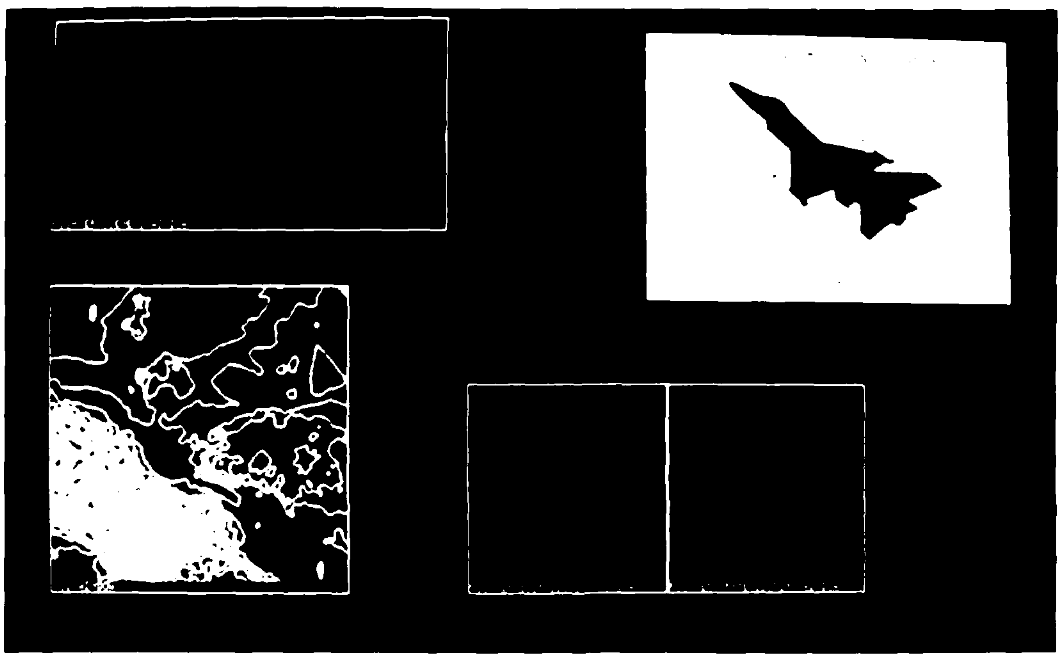


Figure 34. Left Region Magnified

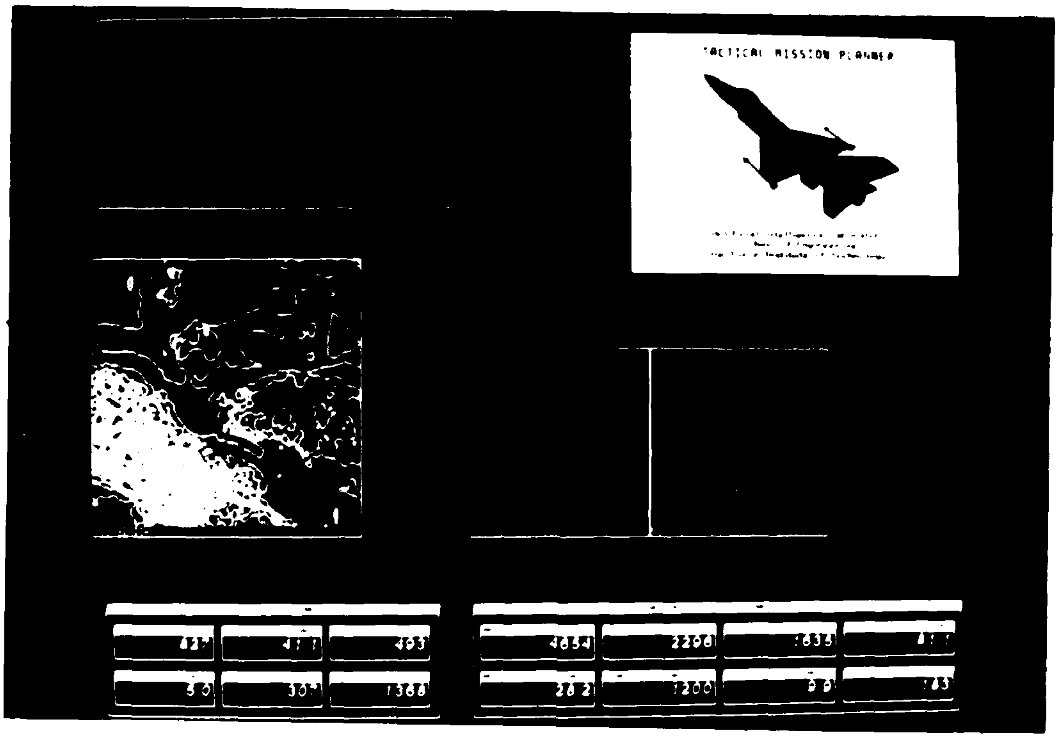


Figure 35. Right Region Magnified

<u>Threat Displays and Mission Impact Analysis</u>

Assuming a mission plan has been prepared, the pilots completed their briefing and are preparing to depart the squadron. An intelligence update located a new SAM site near the target. What is the impact on the planned mission? The intelligence officer, or any squadron pilot, can select, with the middle mouse button, the visual display commands' menu and after selecting 'Clear Display Area,' select 'Place SAM on Map' item, placing the SAM at the reported location. After establishing the look direction or setting the SAM FOV to 360 degrees, the pilot selects Laser Scan LOS (Line Of Sight) (figures 36 and 37).

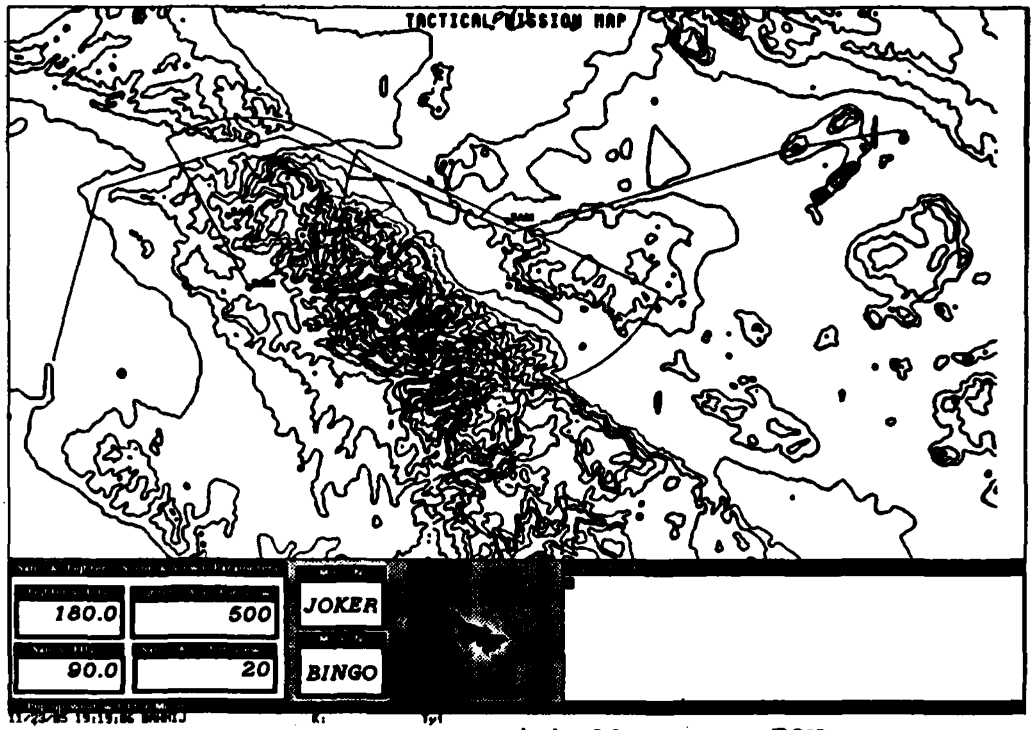


Figure 36. SAM with 90 Degree FOV

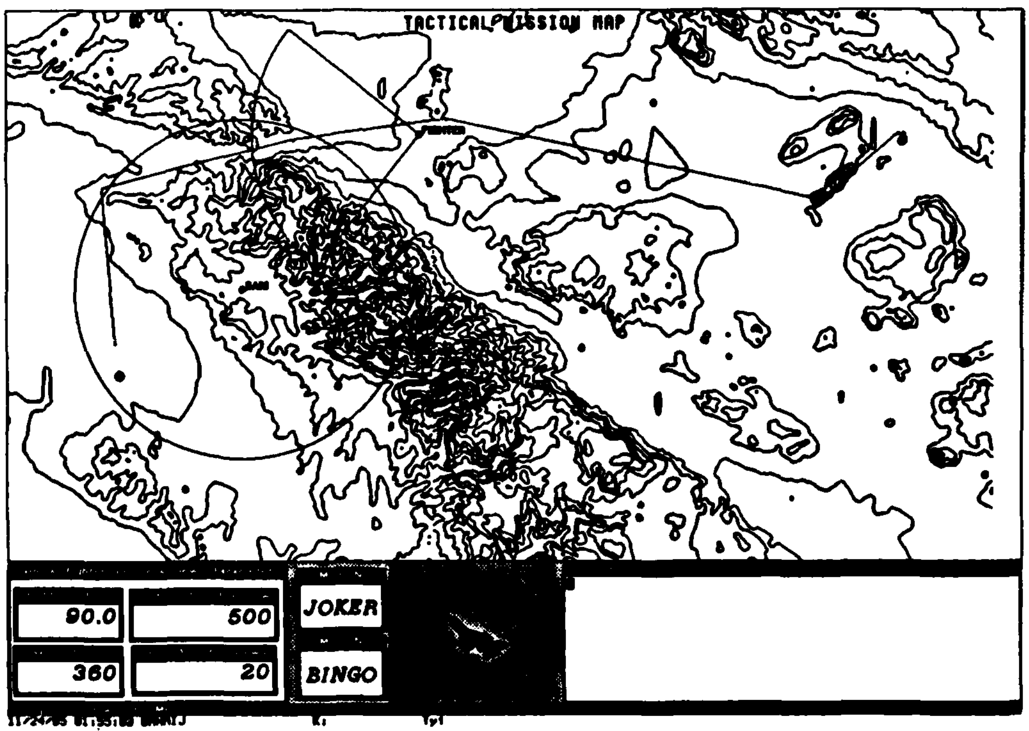


Figure 37. SAM with 360 Degree FOV

The system will now determine if the planned flight trajectory is under SAM threat. The pilots can put on their flight gear, that is, a g-suit, parachute harness, survival vest, and attend to last minute details, such as, ensuring their helmet microphone and speakers, and survival radios are functioning properly.

Conflict Identification. The SAM's radar antenna is set at 20 feet above the ground and the fighter's altitude is planned for 500 feet above the ground level (AGL). These values can be changed via the digiactuator panels on the main screen. The threat assessment phase displays are located on the color monitor. The appropriate portion of the mission terrain map, including any low level legs in that sector, is redrawn on the SC1-WINDOW (see figure 38).

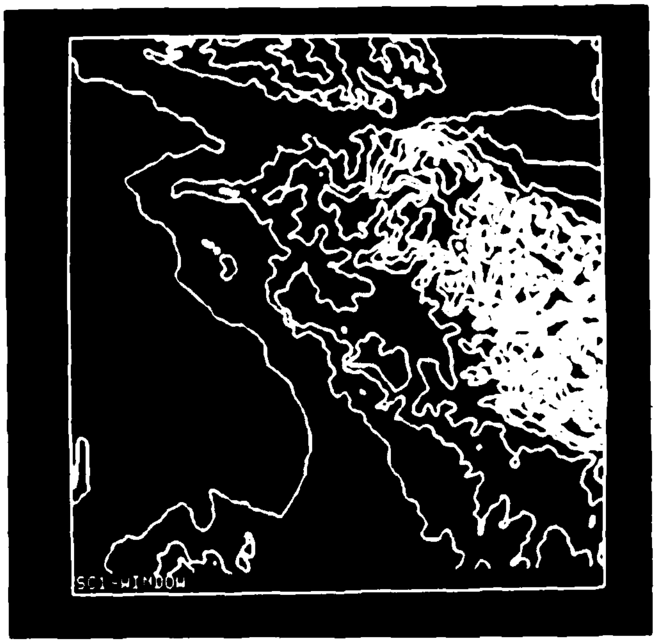


Figure 38. SAM's SC1-WINDOW

The center of the scan is the radar antenna. The scanning function examines each pixel on a particular radial or set of radials, SAM FOV, and determines if it can be seen from the SAM's position and elevation. A pixel, if visible, is colored red. If the pixel is hidden, masked, behind the terrain it will be ignored. The following paragraph will briefly explain the scanning function.

Adding the point elevation (MSL), retrieved from the terrain map data array, to the height of the radar antenna above ground (AGL) determines the absolute elevation of the central looking point. Each point on a given radial is examined to determine if it can be seen. The length of the radial is a function of the range of the specific radar. This range is a slot value of the SAM unit. The function examines the elevation of all the points on the radial. If the elevation of any point between the antenna and the specific point being examined is higher then that point, the point being examined must be hidden behind the taller intermediate point. This calculation continues for each point on the radial until the line of sight visibility of each point is determined. This basic process is also used to produce the terrain profile fighter views. However, for the threat detection variant a crucial step has been added to the algorithm. After the elevation value is retrieved and before the point is processed for visibility, the point (or pixel) is querried if it is located on any navigation

leg. If the point of interest is on a navigation leg the fighter's AGL **ALTITUDE** is added to the point before it is sent to the visibility determination step. This demonstrates the concept of embedding knowledge in the screen map pixels, making the mission map 'smart.' Thus, the threat analysis is truly terrain and threat radar capability dependent. If a point, which is on a navigational leg, is visible (see figure 39) its navigational coordinates and scan radial are stored for possible future use. A message is sent to the leg or a SAM threat monitor/resolver knowledge source (KS), warning of the possible threat detection. These knowledge sources can now determine how to resolve this conflict.

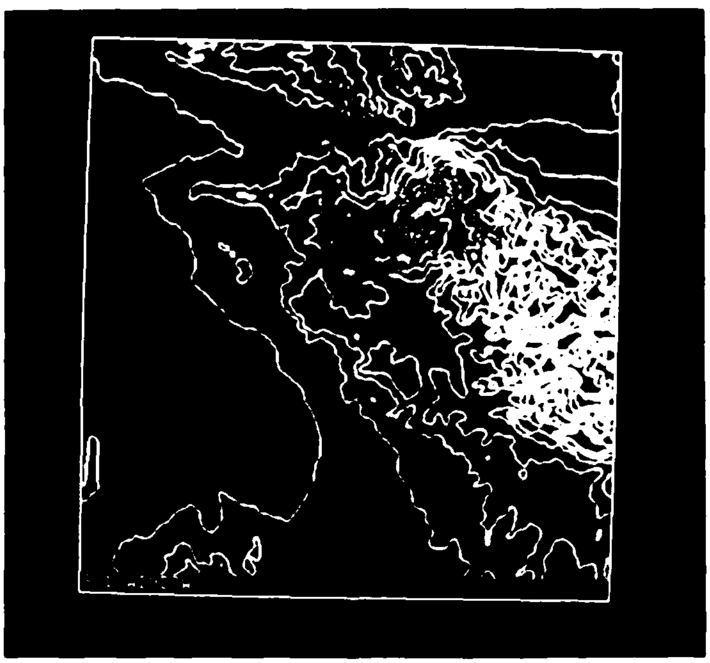


Figure 39. SAM Detects Fighter on Nav Leg

Conflict Resolution. The SAM conflict resolver unit contains procedures and rules, representing the domain expert's approach to resolve a similar conflict. The following scenario presents a possible SAM threat resolution sequence. Being visible at the proposed flying altitude, the KS will determine if the point can be seen on the ground by rescanning the appropriate radials after resetting the fighter's altitude to 0. If the SAM can see the point, indicating a clear unobstructed view, then there is no need to search for an altitude to under fly the radar threat coverage, for one does not exist. If the point is hidden on the ground, the KS will rescan the points on those radials that were previously visible. The scan will sequentially decrement the fighter's altitude by 100 feet until a clear, save, altitude is produced and proposed to the pilot. Scanning only a limited number of radials decreases processing time. If the radar threat cannot be underflown, the route should be moved.

The pilot can select points or legs to move or modify. However, if moving these points violates strict mission constraints the pilot will be forced to fly through the threat. Specific threat domain information transformed into relevant knowledge can better prepare the pilot. The SAM unit contains the specific domain information and the threat resolver KS unit contains the heuristics to convert this information into knowledge. The SAM unit contains the time

required to launch a missile at the fighter from the initial radar target acquisition. The KS has stored the leg segment that was vulnerable. The AFIT unit contains its maximum airspeed value.

Therefore, if the fighter was exposed for 25 seconds at 480 knots and the SAM needed 30 seconds to launch a missile after it acquired the target the fighter could probably fly through the area without having a missile launched at it. If the SAM needed only 20 seconds to launch the fighter would be in trouble at 480 knots. If flying at the maximum speed for its configuration, the fighter can cross the vulnerable segment in less than 20 seconds, again flying past the threat. The pilot would know the exact position when he would be detected and for how long he was exposed. The pilot can deploy, position his flight, typically four fighters, to optimize threat lookout and reaction and minimize detection. He would also know at what point the flight would have to be at maximum speed, as well as, how long they would have to maintain that speed. Flying at maximum speed decreases fuel at a much greater rate, an undesirable predicament. However, having this knowledge keeps the high fuel consumption flight time to a minimum. The flight's situation awareness has been increased by the knowledge presented by the system proposed threat resolution options.

Increased Pilot 'Situation Awareness'

The entire mission planning process focussed the

pilot's attention on the overall mission environment. Since most of the low level computationally intensive tasks were off-loaded to the computer, the pilot has more time to get familiar with the terrain and conduct 'what if' sensitivity analyses. The knowledge acquired through use of this type system better prepares the pilot to fly in this dynamic domain. The system potentially allows more mission level knowledge to be assimilated and the formulation and organization of decision rules which can be accessed in the real-time context of the surface attack domain. The time frame of this research project (approximately ten weeks) did not allow an operational evaluation.

Summary

This chapter described a working tactical mission planning prototype. It was designed to be interactive, exploiting the strengths of both man and machine, to overcome their respective limitations. The added synergistic benefit of increased time to focus the pilots attention on mission level aspects of planning, leading to increased mission situation awareness, is of great value. This last point is the single most important concept a system designer should get from reading this thesis.

Again, it is emphasized the work described here is the first iteration of a functional mission planning prototype designed for use in the squadron, that is, on the ground prior to flight. Approximately three additional months

would be required to upgrade the capabilities of this prototype so it could be placed in selected operational squadrons (Beta test sites) throughout the TAF. Prototype development and technology transfer to operational units are potential projects for the proposed AFIT Center for Strategic Computing and Artificial Intelligence. Further work in the tactical mission planning area will be accomplished in future AFIT theses, described in more detail in the next chapter.

VI. Summary and Conclusions

Introduction

The major contribution of this thesis is the design, implementation, and evaluation a demonstrable interactive ground-based tactical mission planning prototype. This system was 'knowledge engineered' by a domain expert. The work demonstrates that given a properly designed knowledge representation language and programming environment, a domain expert can design a knowledge-based prototype. The resultant prototype would be used to further develop system requirements. There are two major conclusions which may be drawn from this thesis. First, this prototype allows users to easily define system requirements. This must be an iterative process for a complex domain. Secondly, this prototype focuses the pilot's attention on mission level planning tasks. By off-loading the time consuming laborious tasks, the system reclaims valuable time. The pilot now has more time available for refined mission planning and contingency analysis.

These conclusions are significant in light of the current high interest in developing plausible knowledge-based systems for the single seat fighter. This interest has been generated by DARPA, sponsored by the Pilot's Associate Office and perpetuated by industry. These systems must be designed to meet the realistic operational needs of the fighter community, who are the intended end users. The

deployment and use of several such prototypes in operational squadrons can, within six months, help define realistic mission planning system requirements. The outcome of this proposed system definition process can further benefit industry, currently involved in designing systems for the Pilot's Associate effort, by presenting the domain users' perspective.

This chapter will examine the role of the domain expert in system development, highlight the major conclusions of this research, and propose future research and extensions to the current prototype.

Role of Domain Expert

The traditional role of the domain expert in commercial system design needs to be re-evaluated. The current shallow and biased view of knowledge engineering, held by knowledge engineering experts and graphically depicted in Waterman's most recent book, portrays the knowledge engineer as the sole link to the 'Expert System' (Waterman, 1986:5,8). The domain expert's function is reduced to answering domain related questions in an attempt to give a knowledge engineer a 'CLUE.' The results of such approaches to system design are costly systems which fail to meet user needs.

Figure 40 displays the author's view of the domain expert's role. The expert's involvement in system design and development is mandatory for producing successful commercial delivery systems. The domain expert, trained to

program in a knowledge representation language, works in concert with the knowledge engineering expert and consulting with subtopic area experts to develop a viable product.

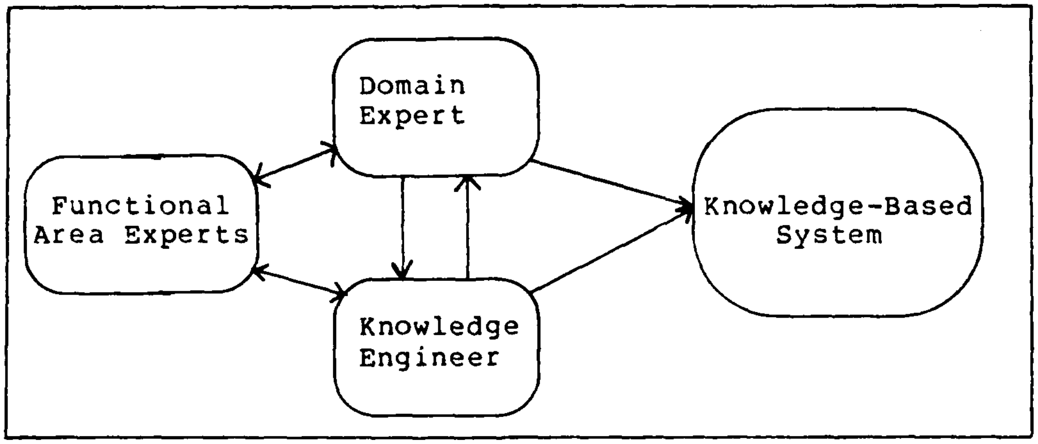


Figure 40. Realistic Domain Expert's Role

The key observation is that given the proper tools, it is more cost effective and less time consuming to train a motivated fighter pilot to be a computer knowledge engineer, than to teach a knowledge engineer to be a fighter instructor pilot. Not only is this true, but the design more quickly converges to an acceptable solution. The author's current assignment is an example of the Air Force's attempt to train domain experts to develop systems which will meet operational requirements, by means of the graduate computer engineering program at the Air Force Institute of Technology. Through the course of this research it was observed a knowledge-based language was required to allow the domain expert to support rapid prototyping.

The Symbolics Zetalisp system coupled with a knowledge-

based system engineering tool, such as, KEE, create an environment conducive to exploratory programming. As was shown in chapter five, generic tools often need to be modified and tailored to meet the operational needs of the user. This is and has been an important basic software engineering principle.

Rapid prototyping and easy system modification are essential system ingredients during the problem domain analysis and system requirements definition phase. A working prototype needs to be transferred to the users as quickly as possible. The evolution of this prototype leading to a refined mature prototype will determine realistic user needs and produce accurate delivery system requirements.

System Evaluation by Pilots

This prototype has been demonstrated, including a briefing highlighting the author's approach to designing systems which meet operational needs, to more than a dozen general officers (Army and Air Force, active duty and retired). They unanimously reaffirmed the need for these prototypes and supported the active involvement of domain experts in technology development and transfer. Several generals actively participated in modifying the user interface panels. For instance, one general officer said he wished the joker and bingo constraint violations to be graphically displayed. The suggested modifications were implemented within two hours. Another general officer

requested a flight mission card be generated. Although this was not completed, another officer unfamiliar with the source code identified the appropriate KEE objects, including their slots, and within thirty minutes, wrote a procedure to produce such a card at the end of the planning session. Some changes produced desirable effects, others did not. In all instances the process of implementing the perceived improvements and re-evaluating the resulting system helped define operational system requirements, reflecting the needs of domain users. After a short introduction to this prototype, each general visualized and proposed several extensions, which reflect current operational needs. The acceptance of this prototype and the author's conceptual approach to problem solving was so strong, Major General Brechner, 17th Air Force Commander, USAFE, is arranging a trip for the author and his thesis advisor to fly to Europe, to evaluate current tactical systems and consult on proposed systems. This prototype not only served as a catalyst, generating users' activity in developing future system requirements, but also guided and focused this effort in the appropriate direction.

A second group of eleven pilots evaluating this prototype are assigned to the School of Engineering, Air Force Institute of Technology. Five are faculty members. During the fall term 1985, as a class project for EENG 548, Man-Machine Interface System, six students used and evaluated

this prototype. This evaluation compared the current mission planning system used in operational squadrons with this prototype. The four major conclusions of this analysis follow (Sheehan et al, 1985). Pilot mission situation awareness was improved through the use of the prototype rather than the current system. The information displayed on the parameters and alarm panels provided immediate real-time feedback giving the pilot more control over the planning process rather than input data and wait for results. The prototype permitted the pilot to do 'what-if' contingency planning, without paying the high cost of replanning the entire mission. The prototype's growth potential is unlimited due to the modular functional decomposition of the knowledge base and the capability for domain user enhancements, tailoring the system to specific operational needs.

Further Research and Extensions

This thesis, including the working prototype, has spawned numerous topics requiring further research, as well as, additions or extensions, which lab engineers can easily build. At least four new thesis students will pursue further research under the direction of Captain Cross. A brief description of proposed future research topics and system extensions follow.

Attack Planning Aid. The mission planning phase, starting at the Initial Point (IP), planning the actual

attack on the target, and egressing the immediate target area is the next logical research topic. The same paradigm used in this thesis is directly applicable for this subdomain. The author has already incorporated a skeletal design of that extension in the current knowledge base. KEE class units have been created, along with appropriate slots, containing relevant information and in some instances containing procedures already written. Each target knows which weapons are appropriate to destroy it and also the priority ordering of those weapons. The targets also know the type of weapons delivery events, how the pilot will actually drop his bombs on the target, are required for best possible target destruction. These weapons events contain slots with information and procedures to assess target area weather conditions for possible constraints, proposing the most reasonable attack option and delivery event.

The current prototype and the proposed attack planning prototype would interface at the IP. The mission critical parameters shared at the IP serve as constraints on the entire mission planning effort. This type of problem decomposition is necessary for parallel processing applications. The system would also allow the pilot to specify how standard functions were to be incorporated into an operational flight program (OFP). For example, the pilot would optimize EW jamming performance during preflight planning based on the weapon delivery tactics he chooses to employ.

Flexible/Intelligent Aircraft Weapons' Configurations. An Air Force Officer, assigned to AFWAL Avionics Laboratory, at Wright-Patterson AFB, is currently working on the flexible configuration portion of the prototype. The squadron pilots and more importantly the weapons maintenance personnel would like to have the capability of stating just the number and type of weapons to load on the aircraft, with the system determining the best location to hang the bombs, minimizing drag index and complying with aircraft stability constraints. The system would also compute the desired mission values, such as, weight and drag index. The need for such a system is highlighted in the following scenario. The exact weapons delineated in the ATO are not available on base. It would be advantageous for a system to automatically compute equivalent weapon loads, optimizing selected variables and immediately transmitting these new constraints to the planning party.

Mission/Aircraft System Monitors. Mission impact sensitivity analysis can be incorporated in this prototype. Experts in domain subareas, such as, radar, fire control, flight control, avionics, and electrical systems, can design and develop knowledge sources, that is, modules. These KS would have the knowledge required to determine the impact on overall mission performance given some specific system component degradation or fault. The skeletal units already exist.

Parallel/Co-Processing. Further research will be conducted in domain problem decomposition and applying parallel processing techniques to reduce the time cost associated with a single processor unit. Examination of current distributive architectures and development of new hybrid architectures are some objectives.

Hardcopy of Planning Results. This extension would produce a strip map, depicting the proposed route of flight with all relevant data imprinted on the map periphery. A detailed diagram of the attack(s), from the IP to actual weapons delivery, including timing, speed, heading, and action points also imprinted on the diagrams. This task can be accomplished by a system programmer, not a researcher.

Operational Assessment. This prototype needs to be deployed to several fighter squadrons for operational assessments in two major areas. Experimental studies need to be conducted to determine what extent, if any, properly designed systems can increase pilot situation awareness. Bill Rouse, at Georgia Tech, is actively involved in research in this domain. The Human Resources Laboratory at Wright-Patterson AFB, also conducts similar research.

The second reason for deploying these prototypes, is to quickly define the system requirements for run-time versions. This has to be done quickly, before much money is spent on hardware that may not support operational software

needs. After a demonstration and hands-on mission planning, Major General Todd, Vice Commander of Air University, stated emphatically, "we need to place it in a squadron now!"

The author suggests this prototype be placed in the Springfield Ohio Air National Guard unit, currently flying the single seat A-7 fighter/bomber. Many of these pilots work in various laboratories at Wright-Patterson AFB and could assess the system from both a pilot's and engineer's perspective. The Air Force Reserve unit, based at Wright-Patterson flying the two seat F-4 fighters, is also a cost effective candidate. DMA terrain data for the local flying area can be acquired, providing the impetus for developing an efficient method to convert the information into a usable form. This data will serve as the basis for the contour maps and elevation arrays. The use of array processors will greatly reduce the data conversion process time.

Summary

A knowledge-based approach to system design and implementation produced a working prototype of an interactive ground based tactical mission planning system. The author applied an object-oriented paradigm, incorporating the Symbolics lisp environment and the KEE knowledge engineering tool, producing a knowledge-based language. This language will permit squadron pilots, the end users, to define commercial system requirements. This approach to system design will benefit both pilots, who have to live with these

systems, and engineers, who now have realistic system specifications on which to base their design.

Situation awareness can be increased with properly designed systems. This prototype exploited the strengths of both man and machine to overcome the shortcomings of each, producing a 'win-win' situation. The interactive use of this prototype has the capability to synergistically increase tactical mission situation awareness.

APPENDIX A

List of Frequently Used Abbreviations and Terms

AAA: Anti-Aircraft Artillery

AAR: Air to Air Refueling

ABCCC: Airborne Command, Control, and Communications

AGE: Aircraft Ground Equipment

AOB: Air Order of Battle

ATC: Air Traffic Control

ATO: Air Tasking Order

AWACS: Airborne Warning and Control

BAI: Battlefield Air Interdiction

Bingo: The amount of fuel needed to get from present location to the airfield of intended landing with the predefined reserve fuel. Bingo is the "point of no return" fuel amount

CAP: Combat Air Patrol

CAS: Close Air Support

Chaff: Packets of thin metallic strips, released from airplanes, used as a radar decoy. Chaff is one example of ECM

Chattermark: Prebriefed procedures to change radio frequencies, whenever the current frequency is being jammed, made unusable)

Comm Jam: Communications' Jamming (denying use of the radios)

Co-Pilot: No such word in the fighter pilot vocabulary

Dash 1: T.O. (Technical Order) 1F-16A-1. The aircraft flight manual

DCA: Defensive Counter Air

DR: Dead Reckoning (the primary means of navigation, combining time, distance, heading, and map reading to accurately fly to a particular destination)

ECCM: Electronic-Counter Counter Measures

ECM: Electronic Counter Measures

E & E: Escape and Evasion

EW: Electronic Warfare

EW Radar: Early Warning Radar (enemy long range radars)

FAC: Forward Air Control (a USAF pilot, assigned to an ARMY unit. Typically flying slower propeller driven planes, the FAC controls and directs the air support for the ground troops. FACs are used for CAS missions)

FEBA: Forward Edge of the Battle Area

FLOT: Forward Line Of Troops (replaces the term FEBA)

Form 70: Mission flight card, carried and used inflight, contains essential numerical mission information (i.e. turnpoint coordinates, times, distances, headings, speeds, altitudes, fuels, target info)

Frag: Fragmentary Order (part of the ATO)

Freq: Radio Frequency

GCI: Ground Controlled Intercept (GCI typically refers to the ground radar personnel/facilities)

INT: Interdiction

IFE: In-Flight Emergency

IFF/SIF: Identification Friend or Foe / Selective identification feature (term SIF is seldom used now)

IP: Initial Point (Excluding the target, the IP is the most important mission point. The IP, located very near the target, has to be easily recognized, both visually and with radar. The IP is used to update all the navigation and weapon systems. The IP starts the attack phase of the mission and is paramount in establish mission navigational orientation and situation awareness

IP: Instructor Pilot

JNC: Jet Navigation Chart. Very large scale (1:2,000,000) theater mission map

JOG: Joint Operations Graphic (Air). Detailed map (scale 1:250,000) used for target area attack planning

Joker: Fuel amount, typically set 500 pounds above bingo fuel, is used to warn the pilot that bingo fuel will soon the important mission factor. If the pilot is currently engaged with the enemy (either air or ground), he must now plan to disengage and start heading home. Many planes were lost in previous conflicts, when the pilots fought past bingo fuel

JMEM: Joint Munitions Effectiveness Manual (this manual defines the amount and type of resources to destroy any target)

LLTN: Low Level Tactical Navigation (a phase of the mission, sometimes referred to as 'Ingress')

LLTR: Low Level Tactical Route

MIKE Plan: A specific Mission plan

OCA: Offensive Counter Air

ONC: Operational Navigational Chart. (1:1,000,000), large scale mission map

Ops: Operations. (refers to the senior management of any fighter squadron, i.e. the Squadron Commander, Operations Officer and the Assistant Operations Officers)

ROE: Rules Of Engagement (may be classified as political considerations or tactical fundamentals. Political restrictions on our tactics are realities that must be complied with in mission planning and execution).

R/T: Radio Transmissions (Pilots' use of the radios)

RWR: Radar Warning Receiver

Safe Area: Geographical regions, located behind enemy lines, which have been selected as the best areas to support E & E efforts, in case the pilot has to eject

SAM: Surface to Air Missile

SAR: Search And Rescue

Sortie: A count of actual aircraft flights. (One flight by a single aircraft, from takeoff to landing, is one sortie)

SPINS: SPecial INStructions

TOT: Time On Target

TPC: Tactical Pilotage Chart. Large scale (1:500,000) mission map. Typically used as the mission overview map

Weasels: Wild Weasels. Presently F-4G's, specially electronically equipped fighters. Their mission is Defense Suppression (DS), locate and destroy SAMs, or at least prevent the SAM sites from launching on friendly strike aircrafts. Weasels are typically part of every strike package

APPENDIX B

Combat Mission Planning Checklist

I. Collect Information

A. Current Readiness Posture (alert state)

B. Frag --

1. mission number
2. target or mission objective
3. force structure
4. ordnance
5. routing factors:
    a. AAR
    b. rendezvous points
    c. CAP points
    d. mandatory penetration points, altitudes
    e. chaff corridors
6. TOT/vulnerability period
7. frequencies
8. IFF procedures
9. coordination/ points of contact

C. Read File/ Spins (ROE)

D. Intelligence

1. home base threats
2. location of FLOT/FEBA
3. location of suspected/known SAMs and AAA
4. fighter threat, GCI capability
5. comm jam
6. E & E procedures (SAFE areas)
7. location of friendlies
8. enemy capabilities:
    a. readiness
    b. aggressiveness
    c. order-of-battle, tactics

E. Your Resources

1. aircraft -- number and configurations
2. munitions and fuzes
3. pilots:
    a. number, experience, proficiency
    b. crew rest
4. time available for planning

5. ground support:
    a. personnel, AGE
    b. runways (barriers)
    c. ATC facilities
6. GCI/ AWACS

F. Mission Environment

1. day/night
2. weather:
    a. cloud cover
    b. visibility (haze)
    c. sun angle (shadows)
    d. contrails

3. terrain:
    a. type
    b. ground cover

G. Deconflict with Other Forces

H. Firm Up Timing at Control Points (takeoff, AAR)

II. Create Administrative Plan

A. Ground ops

1. life support considerations (exposure suit)
2. times -- brief, step, start, takeoff
3. taxi/marshalling (comm out?)
4. aborts/spares

B. Airborne ops

1. takeoff sequence (takeoff data, weight)
2. joinup
3. departure/ recovery:
    a. routing
    b. airspeeds
    c. altitudes
    d. formations
    e. systems checks (switches)
    f. R/T
    g. threats and counters

4. rendezvous with escort
5. AAR data (pre/post strike)
6. joker/bingo fuels (for target, AAR, alternate fields)

7. Go/No-Go decisions:
   a. systems
   b. forces
   c. weather
8. code words (fuels, abort, IFE, chattermark, freq)
9. inflight reports
10. recall/divert procedures
11. air aborts
12. emergency fields
13. SAR

III. Air-To-Surface Tactical Plan

A. Target Destruction

1. target vulnerabilities
2. appropriate munitions, fuzes
   a. types and number (JMEM)
   b. fuze settings
3. impact angle and spacing
4. delivery mode
5. attack axis
6. flight frag deconfliction
7. weaponeering (complete worksheet to get release altitude that will insure fuze arming and safe escape)
8. delivery parameters (complete that worksheet)
9. backup delivery, parameters

B. Target Area Tactics

1. select definable IP
2. IP-to-target routing (threat avoidance,DR)
3. aimpoints (first impacts downwind)
4. attack plan:
   a. airspeeds (use of burner)
   b. formations
   c. sequence, timing
5. delivery considerations:
   a. employment limits (dash 1)
   b. techniques
6. flight reform after delivery
   a. airspeed
   b. maneuvering, calls
   c. visual pickup point
7. timing constraints
8. use of support forces
9. threats --counter, ECM/ECCM

10. contingency plans:
    a. missed IP or missed target (reattack)
    b. battle damage
    c. no release (dump target, higher fuel flows)

C. Ingress/Egress Tactics
   1. routing (deconflict from other forces)
   2. altitudes (deconflict from other forces)
   3. airspeeds (timing)
   4. formations
   5. responsibilities:
      a. navigation
      b. formation
      c. visual, radar lookout
      d. R/T (discipline)
   6. counters/reactions:
      a. comm jam (chattermark freq)
      b. threats:
         1. flight maneuvering
         2. use of RWR, ECM
         3. defensive ordnance (switches)

      c. store limitations:
         1. carriage
         2. jettison

IV. Coordinate With:

A. Base Units--

1. maintenance and weapons
2. intell
3. weather (brief)
4. air base defenses
5. command post
6. ATC facilities

B. Off-Base Units--

1. GCI/AWACS
2. tankers
3. escort
4. supporting units (weasels, FAC)
5. SAR forces

V. Assemble Pilots and Complete Mission Planning

A. Assign Duties to Accomplish:

1. map preparation and weaponeering
2. Form 70 or equivalent
3. photo study
4. E & E materials
5. authenticators

B. Allow Adequate Time for Route and Target Area Study

C. "What if" the plan--

1. aborts, IFEs
2. weather
3. takeoff delays (single runway)
4. late or no-shows:
    a. tanker
    b. AWACS
    c. escort
    d. CAP
    e. FAC
5. comm out plan

VI. Briefing

VII. Post-Mission Duties

APPENDIX C

Flight Mission Planning Duties

I. Mission Planning Duties : Flight Lead

As A Flight:

1. Receive frag.

2. Receive initial weather brief.

3. Receive intell brief.

Individual Duties:

1. Plot target area threats and select IP (1:50,000 chart).

2. Consider target vulnerability and available munitions in choosing attack axis and delivery parameters.

3. Plan target area tactics.

4. Pass delivery parameter requirements to #3 for weaponeering.

5. Plan egress from target to exit point.

6. Coordinate with #2 and #4 on IP and turn points to be used from the FEBA to the target and target to exit point.

7. Reproduce four (4) JOGs with IP-to-target-to-exit point. Include all navigational information and action points.

With #3 :

1. Receive final weather update.

2. Receive intell update.

3. Present plan to ops representative for approval.

II. Mission Planning Duties : #2 (Wingman)

As A Flight:

1. Receive frag.

2. Receive initial weather brief.

3. Receive intell brief.

Individual Duties:

1. Organize planning equipment (templates, plotters, etc).

2. Work with #4: plot points read off by #4 from the Mike plan overlay (from start point to FEBA).

3. Measure headings and distances. Annotate on master map and read to #4.

4. Copy times and fuel flows computed by #4 onto master map.

5. Plot threats on map using threat overlay.

6. Coordinate with #1 and #4 to get points selected from FEBA to target and back to exit point. Plot these points on the map and accomplish steps 2, 3, and 4.

7. Plot return from exit point to recovery field. Accomplish steps 2, 3, and 4.

8. Compute MEA/SAA for low-level legs with #4's assistance.

9. Duplicate three (3) TPC's from the master map. Insure time and distance tick marks are included. Also circle the point used to determine MEA/SAA. #4 will assist.

10. Duplicate three (3) JNCs with #4's assistance. The JNCs will provide routing from home base to low-level start point and return from the LLTR exit point. Annotate emergency fields, MEAs.

11. Mark all maps and cards with the appropriate security level. Handle as classified after marking.

III. Mission Planning Duties : #3 (Alternate Flight Lead)

As A Flight:

1. Receive the frag.
2. Receive initial weather brief.
3. Receive intell brief.

Individual Duties:

1. Obtain delivery conditions from #1 and fill out weaponeering worksheets to get delivery parameters, or use standard parameters.
2. Act as planning coordinator:
    a. monitor progress in regard to time line.
    b. assist wherever needed.
3. Coordinate with intell for:
    a. ECM pod settings for anticipated threats.
    b. chaff/flare settings.
    c. threat reactions.
    d. review of threat slides and book.
    e. Mike Plan procedures pertinent to the mission.
    f. IFF/SIF cards.
    g. E & E materials: kits, procedures and SAFE areas.
4. Compute takeoff data.
5. Review alternate airfield data.
6. Accompany #1 to update briefings and ops briefing.
7. Brief threat and Mike procedures to the flight.

II. Mission Planning Duties : #4 (Wingman)

As A Flight:

1. Receive frag.

2. Receive initial weather brief.

3. Receive intell brief.

Individual Duties:

1. Obtain required charts and lineup cards.

2. Work with #2: Read him the Mike Plan points as he plots them.

   a. copy turn point coordinates on master lineup card.

   b. copy headings and distances as #2 measures them.

3. Use planning sheets to compute leg times and fuels.

   a. write those numbers on lineup card and give to #2 to copy onto master map.

4. Coordinate with #1 and #2 to get turn points from FEBA to target to exit point. Accomplish steps 2 and 3 for those points.

5. Obtain return flight from exit point to recovery. Accomplish steps 2 and 3 for those legs also.

6. Assist #2 in computing MEA/SAAs.

7. Duplicate three (3) lineup cards from the master. Also assist #2 in duplicating three (3) TPCs and JNCs. Insure all time and distance ticks are correct.

8. Refigure times and fuels from takeoff to low-level start point, along low-level, and from LLTR exit point to recovery base.

Bibliography

AFWAL (Air Force Wright Aeronautical Laboratories). "Pilot's Associate Program, Executive Summary and Statement of Work Synopsis," Wright-Patterson AFB OH, 31 Jan 1985.

Aho, Alfred V. et al. The Design and Analysis of Computer Algorithms. Reading, Massachusetts: Addison-Weslsy Publishing Company, 1974.

Barr, Avron and Edward A. Feigenbaun. The Handbook of Artificial Intelligence, Volume 1. Los Altos, California: William Kaufmann, Inc., 1981a.

-----. The Handbook of Artificial Intelligence, Volume 2. Los Altos, California: William Kaufmann, Inc., 1981b.

Benoit, J. W. et al. KBS: An Expert Planning System for Crisis Response. RADC-TR-82-217, Contract F19628-82-C-0001. Mitre Corporation, Mclean, Virginia, August, 1982, (AD-A121 445).

Callero, Monti et al. TATR: A Prototype Expert System for Tactical Air Targeting. R-3096-ARPA, Contract MDA903-82-C-0061. Rand Corporation, Santa Monica, California, August, 1984, (AD-A146 792).

Chandrasekaran, B. "Expert Systems: Matching Techniques to Tasks," Artificial Intelligence Applications for Business, edited by Walter Reitman. Norwood, NJ: Ablex Publishing Corporation, 1984.

Charniak, Eugene et al. Artificial Intelligence Programming. Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc., 1980.

Cohen, Paul R. and Edward A. Feigenbaun. The Handbook of Artificial Intelligence, Volume 3. Los Altos, California: William Kaufmann, Inc., 1982.

DARPA (Defense Advanced Research Projects Agency). "Strategic Computing," DARPA, Washington DC, 28 October 1983, (AD-A141 982).

Davis, Randall. "Expert Systems: Where Are We? And Where Do We Go From Here?," The AI Magazine, 3: 3-22 (Spring 1982).

Doyle, Jon. "Expert Systems and the "Myth" of Symbolic Reasoning," IEEE Transactions on Software Engineering, SE-11: 1386-1390 (November 1985).

Draper, Stephen W. and Donald A. Norman. "Software Engineering for User Interfaces," IEEE Transactions on Software Engineering, SE-11: 252-258 (March 1985).

Engelman, Carl et al. "KNOBS: An Experimental Knowledge Based Tactical Air Mission Planning System and A Rule Based Aircraft Identification Simulation Facility," Proceedings of the Sixth International Joint Conference on Artificial Intelligence. 247-249. Tokyo, Japan, 1979.

-----. "Interactive Frame Instantiation," Proceedings of the National Conference on Artificial Intelligence. 184-186. Stanford University, 1980.

-----. "Knobs: An Intergrated AI Interactive Planning Architecture," AIAA Computers In Aerospace IV Conference. 450-462. AIAA Press, New York, 1983.

Fikes, Richard and Tom Kehler, "The Role of Frame-Based Representation in Reasoning," Communications of the ACM, 28: 904-920 (September 1985).

Fox, Mark S. Constraint Directed Search: A Case Study of Job-Shop Scheduling. PhD dissertation. Computer Science Department, Carnegie-Mellon University, Pittsburgh PA, 1983.

Harmon, Paul and David King. Expert Systems. New York: John Wiley and Sons, Inc., 1985.

Hayes-Roth, Frederick et al. Building Expert Systems. Reading, Massachusetts: Addison-Wesley Publishing, 1983.

-----. "Knowledge-Based Expert Systems," Computer, 17: 263-273 (October 1984)a.

-----. "The Knowledge-Based Expert System: A Tutorial," Computer, 17: 11-28 (September 1984)b.

-----. "Rule-Based System," Communications of the ACM, 28: 921-932 (September 1985).

Hofstadter, Douglas R. GODEL, ESCHER, BACH: an Eternal Golden Braid. New York: Vintage Books, 1979.

Jardine, T.J. and S.T. Shebs. "Knowledge Representation in Automated Mission Planning," Proceedings of the International Joint Conference on Artificial Intelligence, 1:9-14 (1983).

Klahr, Philip and William S. Faught. "Knowledge-Based Simulation," Proceedings of the National Conference on Artificial Intelligence. (1980).

Klahr, Philip et al. SWIRL: Simulating Warfare in the ROSS Language. Rand Corporation Note N-1885-AF, September 1982.

Madnick, Stuart E. and John J. Donovan. Operating Systems. New York: McGraw-Hill Book Company, 1974.

McArthur, David and Philip Klahr. The Ross Language Manual. Rand Corporation Note N-1854-AF, September 1982 (AD-A121 494).

Mitchiner, John L and Laurence R. Phillips. Researchers, Exploratory Systems Division III. Personal discussion. Sandia National Laboratories, Albuquerque, NM, 14 and 15 August 1985.

Mostow, Jack. "Forward: What is AI? And What Does It Have to Do with Software Engineering?," IEEE Transactions on Software Engineering, SE-11: 1253-1256 (November 1985).

Nilsson, Nils J. Principles of Artificial Intelligence. Palo Alto, California: Tioga Publishing Company, 1980.

Norman, Douglas O. Reasoning in Real-Time for the Pilot Associate: An Examination of a Model BAsed Approach to Reasoning in Real-Time for Artificial Intelligence Systems Using a Distributed Architecture. MS Thesis, to be published, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1985.

Parnas, David Lorge et al. "The Modular Structure of Complex Systems," IEEE Transactions on Software Engineering, SE-11: 259-266 (March 1985).

Rasmussen, Jens. "Skills, Rules, And Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models," IEEE Transactions on Systems, Man, and Cybernetics, SMC-13: 257-266 (May/June 1983).

Rich, Elaine. Artificial Intelligence. New York: McGraw-Hill Book Company, 1983.

-----. "The Gradual Expansion of Artificial Intelligence," Computer, 17: 4-12 (May 1984).

Sacerdoti, Earl D. "Problem Solving Tactics," Proceedings of the Sixth International Joint Conference on Artificial Intelligence. 1077-1085. Tokyo, Japan, 1979.

-----. "Overview of Expert Systems Tools." Keynote presentation to the Aerospace Applications of Artificial Intelligence Conference. Dayton OH, 19 September 1985.

-----. "Robot Eye, and ROI: Technology Transformation Versus Tecnology Transfer," The AI Magazine, 6: 204-209 (Fall 1985)b.

Santilli, Lt Col Stan R. "Information Processing and You," Tac Attack, 25: 6-11, Department of the Air Force. (June 1985)

Sheehan, Kevin et al. An Application of Artificial Intelligence: The Tactical Mission Planner. Unpublished class project report for EENG 548, Man-Machine Interface System. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH, 2 December 1985.

Shiel, B. A. "Power Tools for Programmers," Interactive Programming Environments, edited by David R. Barstow et al. New York: McGraw-Hill Book Company, 1984.

Stefik, Mark. "An Examination of a Frame-Structured Representation System," Proceedings of the Sixth International Joint Conference on Artificial Intelligence. 845-852. Tokyo, Japan, 1979.

-----. Planning with Constraints. PhD dissertation. Computer Science Department, Stanford University, Stanford CA, January 1980 (STAN-CS-80-784).

-----. "Planning With Constraints," Artificial Intelligence, 16: 111-139 (May 1981)a.

-----. "Planning and Meta-Planning," Artificial Intelligence, 16: 141-169 (May 1981)b.

TACM 2-1 (Tactical Air Command Manual). Langley AFB VA: HQ TAC, 15 April 1978.

Taylor, Edward C. "Developing a Knowledge Engineering Capability in the TRW Defense Systems Group," The AI Magazine, 6: 58-63 (Summer 1985).

Waddell, Major Don. "Situtaion Awareness," USAF Fighter Weapons Review, :3-5, Department of the Air Force. (Winter 1979).

Waterman, Donald A. A Guide to Expert Systems. Reading, Massachusetts: Addison-Weslsy Publishing Company, 1986.

Wilensky, Robert. LISPcraft. New York: W. W. Norton and Company, 1984.

Wilkins, D.E. "Domain-independent Planning: Representation and Plan Generation," Artificial Intelligence, 22: 269-301 (April 1984).

Winston, Patrick Henry. Artificial Intelligence (Second Edition). Reading, Massachusetts: Addison-Weslsy Publishing Company, 1984.

Winston, Patrick Henry. and Berthold Klaus Paul Horn. Lisp (Second Edition). Reading, Massachusetts: Addison-Weslsy Publishing Company, 1984.

VITA

Major Robert B. Bahnij was born on 30 March 1950 in Mons, Belgium. He graduated from St. Edward High School, Lakewood, Ohio, in 1968. He then attended Cleveland State University, from which he graduated in 1973 with a Bachelor of Science degree in Mathematics. After receiving a commission in the USAF through the OTS program, he attended Undergraduate Pilot Training at Vance AFB, Oklahoma, graduating in April 1975. After completing F-4 training at Luke AFB, Arizona, in March 1976, he was assigned to Kunsan AB, Korea. Completing his remote tour in May 1977, he was assigned to Clark AB, Philippines, were he served as instructor pilot and flight examiner for the 3d Tactical Fighter Wing. In April 1980 he returned to Luke AFB as an F-104 flight commander, instructor pilot, and flight examiner, training German Air Force and Navy pilots. While on active duty he earned the Master of Science in Systems Management degree from the University of Southern California, in January 1981. In August 1982 he was selected as initial cadre to convert the 58th Tactical Training Wing at Luke AFB to the F-16, serving as instructor pilot and wing small computer manager. In May 1984 he entered the graduate computer engineering program at the School of Engineering, Air Force Institute of Technology.

Permanent address: 6710 Ackley Rd.
Parma, Ohio, 44129

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A163944

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/GCS/ENG/85D-1

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
School of Engineering

6b. OFFICE SYMBOL *(If applicable)*
AFIT/EN

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State and ZIP Code)*
Air Force Institute of Technology
Wright-Patterson AFB, Ohio 45433

7b. ADDRESS *(City, State and ZIP Code)*

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
Pilot's Associate Office

8b. OFFICE SYMBOL *(If applicable)*
AFWAL/CCU

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State and ZIP Code)*
Wright-Patterson AFB, Ohio 45433

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.

11. TITLE *(Include Security Classification)*
See Box 19

12. PERSONAL AUTHOR(S)
Robert L. Bahnij, Major, USAF

13a. TYPE OF REPORT
MS Thesis

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT *(Yr., Mo., Day)*
1985 December

15. PAGE COUNT
146

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.
06	04	
09	02	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Artificial Intelligence Expert Systems
Tactical Mission Planning Planning

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Title: A FIGHTER PILOT'S INTELLIGENT AIDE FOR TACTICAL MISSION PLANNING

Thesis Advisor: Stephen E. Cross, Captain, USAF

Approved for public release: IAW AFR 190-1.
LYNN E. WOLAVER 16 JAN 86
Dean for Research and Professional Development
Air Force Institute of Technology (AFSC)
Wright-Patterson AFB OH 45433

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED XX SAME AS RPT. ☐ DTIC USERS ☐

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Stephen E. Cross, Captain, USAF

22b. TELEPHONE NUMBER *(Include Area Code)*
(513) 255-3576

22c. OFFICE SYMBOL
AFIT/ENG

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

Abstract

A working tactical mission planning prototype is described that automates many of the labor intensive, computationally demanding tasks now associated with tactical mission planning. This prototype focuses the pilot's attention on the higher level aspects of the mission, such as, contingency exploration, simultaneously generating the immediate product, a refined mission plan. It also exploits the strengths of both man and machine to overcome the shortcomings of each, producing a 'win-win' situation. The interactive use of this prototype has the capability to synergistically increase tactical mission situation awareness, on which the pilot will base actual in-flight critical decisions.

The present approach to tactical mission planning has several disadvantages. The pilot must concentrate on isolated subtasks. For instance, he must manually determine mission relevant navigational coordinates from maps. He must then type the coordinates into a hand-held calculator or the squadron's PC to determine critical parameters, such as, leg length and fuel used. The plan is refined iteratively. Artificial intelligence techniques can off-load many of these low level tasks and help the pilot deal with mission complexities. This not only "takes the drudgery" out of mission planning, it improves the pilot's overall mission situation awareness.

This prototype knowledge-based system, designed and implemented by a fighter instructor pilot, overcomes present disadvantages and provides several new capabilities. Examples of new capabilities include: identification and proposed resolution of constraint violations, such as, computer generated advice on threat avoidance options and pilot specification of three dimensional terrain profiles of proposed flight paths.

The research demonstrates that a knowledge-based programming language facilitates system design by domain experts. This language will permit squadron pilots, the end users, to define commercial system requirements. The thesis will describe this system and discuss a preliminary evaluation by Air Force pilots.

END

FILMED

3-86

DTIC